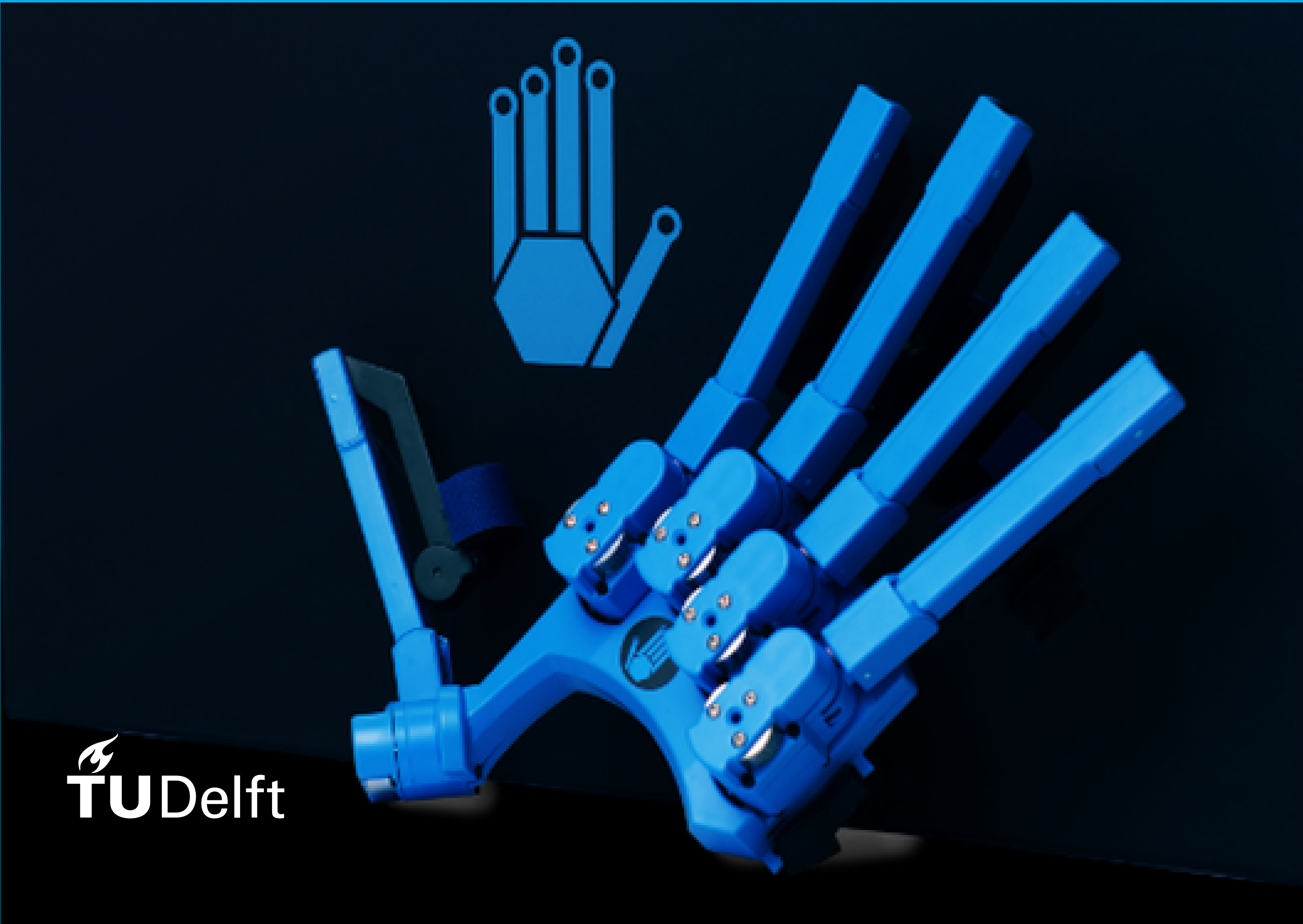
# SoftGlove Finger Vibrotactile Feedback

S.A. Brackenhoff & D.A.M. Koene

## Abstract

In this report the finger vibrotactile subsystem of the SoftGlove wearable haptic feedback glove is discussed. This system is an upgrade over the vibrotactile feedback that the current SenseGlove haptic feedback glove provides. The designed system consists of a linear resonant actuator (LRA) on each finger, which is controlled by a dedicated LRA driver over $I^2C$. To overcome addressing limitations of the drivers, an $I^2C$ switch is used to be able to communicate with each driver individually. The system is characterized both in terms of latency and vibration strength. The peak-to-peak vibration strength is measured to be 1.44 G and a 10 % to 90 % rise and fall time of 44 ms from rest to the maximum vibration strength. Furthermore, the system is integrated with the other subsystems of the SoftGlove on a PCB, the reports regarding these subsystems can be found in [1] and [2].



TU Delft

# SoftGlove

## Finger Vibrotactile Feedback

by

S.A. Brackenhoff & D.A.M. Koene

to obtain the degree of Bachelor of Science
at the Delft University of Technology,
to be defended on June 28, 2019

| | | |
|---|---|---|
| Authors: | S.A. Brackenhoff | D.A.M. Koene |
| Student numbers: | 4619005 | 4593561 |
| Project duration: | April 23, 2019 – July 5, 2019 | |
| Thesis committee: | Dr. M. Spirito | TU Delft, chair |
| | Dr. ir. C.J.M. Verhoeven | TU Delft, supervisor |
| | Dr. J. Dong | TU Delft |
| | Ing. C. Lam | SenseGlove, external supervisor |

*This thesis is confidential and cannot be made public until May 1, 2025.*

# Preface

The past two months have been a tremendous learning experience for both of us. After spending years with our nose in books with only EPO projects to touch upon the reality of engineering, this project was a bit of a culture shock. 11 Weeks seemed like a lot of time to get a few motors to vibrate and we were worried whether we would have enough to do. Looking back at that, we can truly and wholeheartedly say that designing two protoypes in the span of the BAP is quite ambitious, especially with the amount of experience we had in electronic design.

This would have been an impossible feat to complete without the help and guidance of our supervisor, dr. ir. C.J.M. Verhoeven and our client and external supervisor Ing. C. Lam, whom we both want to thank for their time, patience and enthusiasm during the project. Furthermore we want to thank SenseGlove for the opportunity to work on this great project. Additionally, we want to thank dr. I.E. Lager for providing an excellent framework for the BAP. We would also like to thank ing. A.M.J. Slats and M. Schumacher for their help with our experiments, setup and providing us with the components we needed. We owe M. Lammers thanks for tailoring the SenseGlove Unity environment to our needs and helping us design the demonstration of our prototype. We also want to thank D. Shor and ir. M. Corten for their help in getting us familiar with the mysterious field of sensitivity of the human skin and their insights in haptic feedback. Finally, we would like to thank Dr. M. Spirito and Dr. J. Dong for the time and energy they put into being part of our thesis committee and Dr. T. Batista Soeiro for being part of our Green Light Assessment committee.

Last but not least: we want to thank our colleagues of the other subgroups for their help, support and most of all for making working on this project a great time.

*S.A. Brackenhoff & D.A.M. Koene*
*Delft, July 25, 2019*

# Contents

# 1 Introduction

Currently the virtual reality market is booming, new head mounted displays are coming out every few months. These displays contain higher resolution screens, a more comfortable fit, better positional tracking and more. One thing that is lacking in these developments, however, is the existence of haptic feedback. Many virtual reality systems currently on the consumer market use standard remote controllers with a rumble motor inside. However, the key factors in virtual reality systems are immersion and comfort. In order to achieve full immersion a higher resolution screen or better positional tracking is important but the absence of feedback on the touch sense limits the immersion. Seeing yourself picking up an object is one thing, feeling the force on your fingers is another, and increases the immersion immensely. The SenseGlove is a product that aims to attain this immersion by creating a wearable glove for haptic feedback. This is currently possible in the form of an exoskeleton hand, which is not as user friendly as a more flexible fabric glove. Therefore, in this thesis project an improvement to the current SenseGlove is developed and discussed.

In the past, virtual simulations (such as those used in computer games) offered feedback only in terms of clicky buttons. This was later improved with the addition of rumble motors, giving haptic feedback by shaking the controller that was being held. These kind of controllers often had a generic shape, which would emulate for example a person or a tool in a simulation or a game. Up to 1991 [3] the simulation or game was shown on a monitor about a meter or more away from the actor. Virtual reality introduced a display that was positioned so close to the eyes that it almost fully fills the field of view of the user. The immersivity of this technology was initially hampered by the fact that you could only move in this environment by using your controller. This causes many issues regarding simulation sickness, as well as breaking the immersion. With the advent of room scale virtual reality, meaning that the position of the actor is tracked within the room, the actor became allowed to physically move around in this virtual environment. However, the only way to perform an action in this virtual environment is still by using a controller. A controller is intrinsically flawed in providing immersive feedback to an arbitrary action since it is often a generic tool that cannot adapt to the environment the actor is interacting with. Contrarily, in the optimal case the hands are completely synchronized to the virtual environment, while also providing physical feedback. Namely, being able to pick up objects or for example feel the sensation of clicking a button. Furthermore, with additional feedback a multitude of tools could be emulated by providing force or other types of feedback to the hands. This would result in a completely universal kind of controller, since most interactions with tools or the world in general involve the use of the hands.

The electronics that have been designed in this project aim to make a step in improving the immersion of VR by providing more and more detailed types of feedback, as well as operating wireless. The wireless connection enables the user to move around the physical, and thus virtual, environment without being constrained by a wire attached to the hands, which is especially useful when the actor needs to turn around often. By having more types of feedback, a more realistic and complex simulation is possible. This increased realism helps to improve the applications of VR, such as training in virtual reality or more realistic design simulations and verifications. Training people using virtual reality for dangerous or rare situations could greatly benefit from better haptic feedback, as people learn movements more easily in this way [4].

The SoftGlove is based on the current SenseGlove design as described in [5] and [6] and some parts of the design is based on the SenseGlove design in order to better correspond to the temporal and monetary limitations of the BAP. Additionally, some design choices are made in order to integrate with the current protocol of the SenseGlove, to ensure backwards compatibility. The design of the finger force feedback remains as is, meaning that the same actuators are used. In order to facilitate the wireless communication an existing protocol, a transmitter and receiver are used. With regard to the vibrotactile feedback, only off-the-shelf actuators will be used and haptic wave forms are not designed.

## 1.1. State-of-the-Art Analysis

The haptic feedback in VR market is quite a new market. However, a few similar products have already been released or are currently being developed. An analysis of the gloves available in the market can be found in [7]. Another excellent review of the development of VR wearables for the hand in the past decade is presented in [8]. A couple of these products are detailed below.

**VRgluv** This glove features force feedback as well as finger tracking. The force feedback works by providing force on multiple points on each finger. Furthermore, it can operate wireless while running from a rechargeable and replaceable battery. It does not feature a form of vibrotactile feedback, however.

**Maestro** This is a wireless textile glove. Even though the glove is made of fabric, it does feature finger force feedback provided by servo motors that are mounted to the forearm of the user. Additionally, it contains sensors to determine the position of the fingers and includes gesture recognition. Furthermore, it communicates wireless and features vibrotactile feedback on the fingertips, but not in the hand palm. The design is quite bulky, however, amounting to a total weight of about 600 grams per glove, which hinders prolonged use.

**HaptX** A different approach is taken by the HaptX glove. While most of gloves make use of electromagnetic actuator to provide feedback this glove employs a pneumatic system. This pneumatic system is utilized to perform finger force feedback as well as haptic feedback, using microfluid actuators. While the pneumatics provide high fidelity feedback it does have the downside of having to use a stationary compressor and being relatively heavy. A price is not yet announced, but the cost will likely be quite high, since the product is aimed at the industrial market and has to include the compressor.

**Current SenseGlove** The current SenseGlove makes use of an exoskeleton to exert force on the fingers. Each exoskeleton finger also integrates an eccentric rotating mass vibration motor for finger vibrotactile feedback. Furthermore, the joints of all fingers are tracked. Currently this glove only operates over USB, but a wireless kit is in development.

## 1.2. Problem Definition

As can be read in the previous section, many haptic feedback solutions are being developed at the moment. This means that the SoftGlove should not simply be a less bulky variation of the current SenseGlove, but should have additional features that make for a more immersive experience. The SoftGlove should therefore have a performance at least equal to that of the SenseGlove. Additionally, the SoftGlove will feature a larger vibration actuator in the hand palm, which can provide stronger vibrations when the entire hand comes in contact with something in the virtual environment. Additionally, the finger vibrotactile feedback is reevaluated in order to minimize latency and maximize possible vibration strength and possible haptic effects.

## 1.3. Thesis Synopsis

In this thesis, the development of the finger vibrotactile feedback is described. This is done by first elaborating on the problem definition by means of a Program of Requirements. After that, a general overview of the design of the full SoftGlove is presented. Then, the finger vibrotactile feedback design is presented bottom-up, by first discussing the selection of the vibration motor type and the way to drive it and then the system integration. The chosen solutions are then characterized in terms of latency, vibration strength, power consumption and versatility. This component selection and characterization is then applied to a proof of concept and two prototypes which will finally be discussed, after which the conclusions are represented and compared to the requirements. It should be noted that Chapters 2 and 3 are shared between the entire SoftGlove group, whereas the other Chapters are unique to the finger vibrotactile feedback subgroup.

# 2 Requirements

This chapter discusses the general requirements that are the result of the assignment from the company SenseGlove. After detailed research the original assignment is changed to the final assignment. The assignments and requirements are a result of collaboration with SenseGlove about the time and practical limitations of the project. The final assignment and requirements will split the complete system in three subsystems. Finally the requirements that are specific for finger vibrotactile feedback will be discussed.

## 2.1. Assignment

The current version of the product uses an exoskeleton. This design limits the capability and the scale of implementation for augmented reality applications. Therefore a soft, fabric version of the old design is an important development. This soft version should have at least similar capabilities to the current exoskeleton glove, with the exception of finger tracking and added vibrotactile feedback in the palm of the hand. The first assignment made by SenseGlove is discussed in Section 2.1.1. After discussions with the company about the project and research on the subject, the constraints did not completely fit the assignment. Therefore the assignment was modified in collaboration with SenseGlove, this assignment is discussed in Section 2.1.2.

### 2.1.1. Original Assignment

The original assignment was to design and realize a semi-flex PCB for the SoftGlove, which integrates per finger force feedback, linear resonant actuators in the fingertips and a Lofelt haptic actuator on the palm of the hand, including firmware, where communication to the PC through USB according to the SenseGlove protocol is possible. As an optional assignment, the glove can be outfitted with a wireless communication link. This assignment can be found in Appendix A.3.2.

### 2.1.2. Final Assignment

After detailed research it was apparent that some changes needed to be made to the assignment. The semi-flex PCB material is rated to bend a maximum amount of five times to make inserting the PCB in a housing easier [9]. It is not made to bend continuously back and forth and is therefore not suited for bending with the movement of the wrist. Another option would be to use a fully flexible PCB. However the design of a fully flexible PCB adds significant complexity to the design process, as described in [10]. Because of this, the use of a rigid PCB is chosen, which can be mounted on the wrist in the form of several modules.

Secondly, there were some concerns about the assignments challenge level as the finger force feedback is already optimized for the current SenseGlove. Therefore it was decided to make the system work with a battery so the product could become entirely wireless. When making the SoftGlove wireless, power supply by a battery is needed which makes the power conversions for the finger force feedback more complicated. However, the SoftGlove must have the ability to be powered via USB at 5 V with a maximum of 4 A. This results in a maximum available power of 20 W.

## 2.2. General Requirements

Based on the final assignment that is discussed in Section 2.1.2, requirements are set that are applicable for the whole system that should be made for the SoftGlove. The requirements can be divided in mandatory requirements, cost factors and stretch goals. All of these are listed below.

#### Mandatory

1. The glove must have per finger force feedback.
2. The glove must have per finger vibrotactile feedback.
3. The glove must have a larger vibrotactile feedback core in the palm of the hand.
4. The glove must support USB-based firmware updates.
5. The glove may not have a power consumption over 20 W.
6. The average latency of the PCB may be no more than 40 ms. How the latency is defined is discussed in Section 3.2.
7. The PCB must have over current protection.
8. The PCB must have over voltage protection.
9. The PCB must have reverse current protection.
10. The glove must stay under 40°C.

#### Cost Factors

1. The latency of the glove should be as low as possible.
2. Extensions of the glove should take up minimal space on the wrist or other parts of the body.
3. The glove should have a minimal power consumption.
4. The feedback placement on the glove should be optimized where the sensitivity of the human skin is highest.
5. The glove should be as durable as possible.
6. The glove should fit a wide audience as comfortably as possible. This means the product should fit both men and women with a range of different sizes of wrists and hands.

#### Stretch Goals

1. The glove would benefit from being compatible with SenseGlove Communication Protocol [11].
2. The glove would benefit from having a wireless communication link.
3. The glove would benefit from using a mobile power source

## 2.3. Subsystems

It is clear the glove has three major feedback methods, finger force feedback, finger vibrotactile feedback and palm vibrotactile feedback. The finger force feedback can hold the fingers back when they are grasping an object in VR, creating the illusion of a solid object. The other two feedback methods are comprised of vibrations of actuators on the hand, creating the feeling of a buzz when touching something in the virtual environment. The finger vibrotactile feedback is comprised of a smaller actuator on each finger, whereas the palm vibrotactile feedback is a larger actuator in the hand palm. Because there are three types of feedback, the complete system is split up in this three subsystems. Based on the complexity of each subsystem, some secondary tasks are divided to the subsystems. An overview of the placement of all feedback subsystems is shown in Fig. 2.1.

### 2.3.1. Finger Force Feedback

The iconic form of feedback from the client is the Finger Force Feedback, allowing people to "grab" or "squeeze" items in a virtual environment, by applying force to the fingers that stops them from moving through a virtual object. This will be done using the actuators provided by SenseGlove. The actuators provide feedback on the top of all fingers, marked in blue, as shown in Fig. 2.1

This subsystem will use the most power and the highest voltage, and will therefore be accountable for designing the power converters.

### 2.3.2. Finger Vibrotactile Feedback

The more subtle, but just as important way the current version of the glove provides feedback is through small actuators that vibrate the fingers. This system allows the user to experience, for example, button clicks and the smoothness of certain surfaces. This design is meant to be an improvement over the vibration motors currently in the SenseGlove. The finger vibrotactile feedback motors will be placed on the intermediate phalanges of the fingers and the proximal phalanx of the thumb, marked in green, as shown in Fig. 2.1.

### 2.3.3. Palm Vibrotactile Feedback

SenseGlove wants to add another way of feedback in their products, and they want it to be the Lofelt actuator based in the palm. This is a sensitive area that can provide general purpose feedback. The Lofelt actuator will be placed in the palm of the hand, marked in red, as shown in Fig. 2.1.

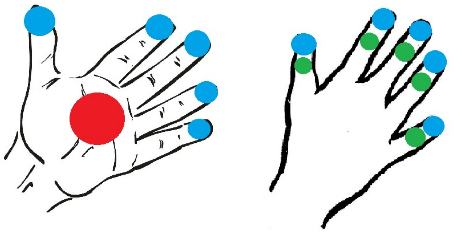


Figure 2.1: Overview of the placement of all subsystems on the hand of the user

## 2.4. Subsystem Requirements

As stated before, the subsystem that will be discussed in this thesis is the finger vibrotactile feedback. This subsystem has some specific requirements beside the general requirements discussed in Section 2.2, these specific requirements are listed below.

- The finger vibrotactile feedback should be at least as good as the feedback currently in the SenseGlove. This concerns the vibration strength and rise and fall time. These are listed in Tab. 4.1.
- The finger vibrotactile feedback should be as versatile as possible, meaning it can quickly switch between vibrations strengths.
- The vibrotactile feedback frequencies of the glove should be in the frequency range where the human skin is at its most sensitive.

# 3
# General Design

Next to the designs of the separate subsystems described in Section 2.3 some general design choices had to be made. These choices are applicable for all subsystems and are discussed in this chapter. The power supply consists of several parts that are split up between the subgroups. First the battery type which is chosen by the Finger Force Feedback group, second the battery charger circuit which is made by the Palm Vibrotactile Feedback group and third the battery protection circuit which is designed by the Finger Vibrotactile Feedback group. Besides the power supply, the microcontroller and programming language were chosen. The way in which the systems cooperate can be found in Fig. 3.2.

## 3.1. Power Supply

As described in the new assignment, which is shown in Section 2.1.2, the goal is to design a wireless glove. For the power supply this means a battery or multiple batteries have to be attached to the SoftGlove or to the human body. As can be seen in the program of requirements, which is shown in Section 2.2, the physical size is a major cost factor. Besides, a smaller system allows the gloves to be compatible for a wider audience, which is also a cost factor. Taking this into account, all considerations and final decisions for the battery type, charger and protection are outlined in this section.

### 3.1.1. Battery Type

Since the SoftGlove is designed for wireless application, a battery has been found that will not constrain the usage of the glove. From the general program of requirements in Section 2.2, some requirements for the battery follow. The battery should be able to deliver a peak power of 20 W and the battery, as an extension of the glove to the wrist, should take up minimal space.

**Types of Batteries** The requirements immediately shorten the list of usable batteries for the application. The used voltages in the system are 3.3 V, 5 V and 24 V, where the 24 V subsystem uses the most power. The highest efficiency will be achieved with a battery input voltage of between 5 V and 24 V. This efficiency is mainly based on the boost from the input voltage to the output voltage of 24 V. When boosting an input voltage lower than 5 V to an output of 24 V, the efficiency of one the boost converter often becomes lower than 75% which is too low to meet the power specifications as described in Section 2.4. This efficiency will be discussed further in the finger force feedback thesis [1]. The second option is to use two boost converters in cascade. However, this uses almost double the space, which is not available. Therefore, the input voltage must be at least 5 V. Furthermore, for practicality and durability the battery needs to be rechargeable. Finally, the battery shape and weight influences comfort of the user of the SoftGlove. Taking all of this in account, five battery types were considered and discussed. Paper [12] was consulted, to further explain the differences between the different batteries. These battery types shown and discussed below. The best battery type is used in the design of the SoftGlove.

- Lead-Acid
- Nickel Cadmium (Ni-Cd)

- Nickel-Metal Hydride (Ni-MH)
- Lithium-ion (Li-ion)
- Lithium-ion Polymer (Li-Po)

**Lead Acid Batteries** Lead Acid Batteries are created as very reliable and low-cost power sources. As disadvantage they have a low energy-to-weight ratio. Because of their big size and high weight in comparison to other battery types, this is not an option for wearable application.

**Nickel Cadmium Batteries** have a couple of useful advantages. For example, they can handle many charge/discharge cycles in comparison to the other types of batteries. On the other hand, there are disadvantages which are so crucial that this type of battery is not chosen for the SoftGlove. Firstly, the presence of the so called 'memory effect': The batteries lose their maximum capacity when they are being recharged after not being fully discharged. Secondly, this type of battery also contains toxic metals and the energy density is not as high as some other battery types. Another disadvantage is that Nickel Cadmium batteries have a cylindrical shape, which is not ideal for efficient usage of the available space on the wrist.

**Nickel-Metal Hydride Batteries** have a higher energy density than Nickel Cadmium batteries but also have the cylindrical shape. The energy density also is not as high as with Lithium batteries. For the same capacity, a bigger and heavier battery is needed. Nickel-Metal Hydride batteries are not effected by the memory effect, which is an advantage. Despite this advantage, the self discharge rate is high and the maintenance to ensure a sufficient lifetime is very difficult. All the disadvantages makes the Nickel-Metal Hydride battery not suitable for usage by a wide and long term audience as for the SoftGlove.

**Lithium-Ion Batteries** are widely used for wearable applications. A disadvantage is that these batteries also have a cylindrical shape. This type of battery is comparable to Lithium-Ion Polymer batteries [13], which have the advantage of a low profile and non-cylindrical shape. Their form factor makes it also easier to attach the batteries to the wrist. Li-Po batteries have a disadvantage of higher price comparing to Lithium-Ion, however these costs small compared to the advantages. Lithium-Ion has a sufficient discharge current for the case of maximal dissipation of 5 A, where maximally 2.5 A can be drawn. Lithium-Polymer generally has even higher discharge rates. Looking at safety differences, Lithium-Polymer is more sensitive compared to Lithium-Ion regarding over voltage and over current while charging and discharging. However, when using reliable and good protection circuits this can be prevented. In Tab. 3.1 the batteries together with their advantages and disadvantages are summarized. Taking all things into consideration, Lithium-Polymer is chosen as the optimal battery type.

Table 3.1: Decision Matrix Battery Type

| Battery type | Advantages | Disadvantages |
|---|---|---|
| **Lead-Acid** | - Non-cylindrical shape<br>- Reliable<br>- Low Cost | - Low energy density<br>- Big size, high weight |
| **Nickel Cadmium** | - Many charge/discharge cycles | - Memory Effect<br>- Toxic metals<br>- Moderate energy density<br>- Cycindrical shape<br>- Self-discharge rate high |
| **Nickel-Metal Hydride** | Similar to Nickel Cadmium but:<br>- Higher specific energy<br>- No toxic Metals<br>- No memory effect | Similar to Nickel Cadmium but:<br>- Less charge/discharge cycles |
| **Lithium-ion** | - High energy density | - Cylindrical shape<br>- Requires specific protection system |
| **Lithium-ion Polymer** | - High energy density<br>- Non-cyclindral shape<br>- Low profile<br>- High discharge rate | - Higher price<br>- Requires specific protection system |

**Integration in Design** Lithium-Polymer batteries have a nominal voltage of 3.7 V. As stated above, it is inefficient to directly convert from this voltage to the 24 V, which is needed for the finger force feedback subsystem. To achieve higher efficiency, two battery cells can be connected in series. This gives a nominal voltage of 7.4 V. The disadvantage of connecting multiple cells in series is the mandatory use of a balancing system between the multiple cells to ensure safety and durability of the cells. From 7.4 V highly efficient boost converters are available that can convert this input voltage to 24 V. Connecting more than two cells in series makes balancing even more difficult and increases size as well. This makes connecting two cells in series the optimal design choice.

Next to choosing the amount of cells, the cell capacity also has to be chosen. This is the amount of energy stored in the batteries. As already said in Chapter 2 the glove must have equal or better specifications than the current model. The wireless kit, that is in development for the current SenseGlove, can last around 30 minutes on maximal power dissipation. To achieve this in the SoftGlove, the maximum power dissipation has to be estimated. Given the nominal battery voltage of 7.4 V, around 2.5 A can be drawn maximally. At this power dissipation the battery must last 30 minutes or more, so a capacity of at least 1250 mAh is needed. A battery is chosen with 1500 mAh capacity, where a maximum continuous current of 4.5 A can be drawn. The size is 66x32x6.5 mm, such that the battery can fit comfortably within the width of most wrists. The weight of two cells is 60 g, not more than the weight of an average watch. These two cells are connected in series to achieve the required input voltage of 7.4 V.

### 3.1.2. Battery Charger

Since the system will be charged over USB, the charger needs to accept an input voltage of 5V. Unfortunately there is currently no IC available with support for boost mode charging, balancing and protection of a 2 cell (2S) lithium-polymer battery. Therefore a separate battery protection and charging IC is used. A single lithium-polymer cell is rated at a maximum of 4.2 V, two cells in series are rated at 8.4 V. Therefore the charger must be able charge the lithium-polymer battery to 8.4 V. The IC used for charging the battery is the BQ25883 from Texas Instruments. This is a 2S boost mode Li-Ion and Li-Po battery charger. It can charge the battery with a maximum current of 2 A. When using the battery as stated in Section 3.1.1 the charging time will be 45 minutes. The final circuit and layout of the charger can be found in Appendix A.1.2 and A.3 respectively.

### 3.1.3. Battery Protection

As stated above lithium polymer batteries need some types of protections. The cells of a Li-Po battery get damaged when they are charged or discharged too far. In case of over discharge the battery will lose some of its capacity and its self-discharge rate will increase. In the case of over charge, the battery might catch fire or even explode. This poses a safety hazard that is not ethically permissible in a consumer product. Because of this, a solid protection circuit is needed. As stated in the section above there is no IC available that can charge, protect and balance a 2S battery. Therefore a separate protection IC is necessary. The battery protection IC that meets all these requirements is the BQ28Z610. While this IC is marketed as a gas gauge, a circuit meant to determine the state of charge of the battery, it also has many protections built in. The IC features over- and under voltage protection, over current protection, short circuit protection and over temperature protection. Apart from these protections it also has the ability to balance a 2S battery. It therefore includes all the desired features that the battery charging circuit lacks. The final circuit and layout can be found in Appendix A.1.3.

Unfortunately the battery protection circuit is untested at time of writing. This is due tot the fact that the footprint of the IC was drawn incorrectly, both in terms of size and orientation. However, this has been rectified for the final prototype and the circuit has been checked multiple times to ensure there are no errors.

## 3.2. Microcontroller

The subsystems of the glove need to be controlled by an microcontroller. Since the desire was to make the system wireless a microcontroller with integrated wireless functionality is ideal. The ESP32 microcontroller was therefore chosen for the prototype as it provides a sufficient amount processing power, storage, IO pins and has integrated Bluetooth and WiFi connectivity. For the final version the ESP32 Pico was selected. The Pico has all the same functionality as the bigger modules, but is a lot smaller with its 7x7 mm QFN package and requires no external components like crystals since they are built into the package. Even though the Pico has Bluetooth and WiFi functionality, it does not have a built-in antenna. Therefore an external antenna has to be used. The Proant 440 was selected, because of its simplicity, small size and good performance.

## 3.3. Programming Language

The chosen ESP32 supports the use of a multitude of programming languages, each with their respective advantages and disadvantages. The programming languages that were considered were Micropython, Arduino and ESP-IDF. The latter is the official development framework based on C provided by the manufacturer of the ESP32. Micropython has the advantage that it is easy to write and especially easy to debug since it is an interpreted programming language. This makes it possible to send commands and read out contents of variables over USB without needing to recompile and upload the code. There are however fairly major disadvantages to this approach. Micropython is slow when compared to Arduino and especially to using the ESP-IDF and it provides little flexibility in regard to for example assigning which pins the I$^2$C bus uses. Another disadvantage is that only a few people in the group have experience with Python and would therefore require some studying of the syntax and behaviour to write proper code. The Arduino programming language benefits from many built-in functions for controlling for example the I$^2$C or SPI bus and it supports the C and C++ languages. However since it is designed to run on a multitude of microcontrollers it features the same flexibility disadvantage as Micropython and is still not as fast as C or C++ code written specifically for the used microcontroller. This is provided by the ESP-IDF, which stands for the Espressif IoT Development Framework. This is the most low level language that has a similar structure to C and C++ and thus provides only limited pre-made functionality, it does, however provide a lot of flexibility and speed. Since a main limiting factor in this project is latency, execution speed of the commands is critical. Furthermore since the whole group has experience in writing C and C++ code from Bachelor courses this would be relatively familiar. Therefore the ESP-IDF was chosen for developing the software that would run on the final prototype. For software development reasons the ESP-IDF code for all subsystems has to integrate with the current SenseGlove communication protocol that is described in [11].

## 3.4. Latency Budget

One of the most immersion breaking parts of virtual reality experiences is latency. It is therefore part of one of the major requirements, namely that the average latency may not be more than 40ms. In order to understand which parts of the design have the highest latency a latency budget was constructed. First of all an estimation was made regarding the various components of the design. After the design and assembly, the actual latencies of the components was measured to check if the estimations were correct. The wireless communication, processing on the microcontroller, the driving of the finger force feedback actuators, the per finger vibrotactile feedback and the Lofelt circuitry were considered in the estimation of the latency budget. The estimated latency budget can be seen in Tab. 3.2. The latencies of the different subsystems have been measured and can be found in Tab. 3.3. The latency of the finger force feedback stays the same because it is based on the known switching delay and rise time of the MOSFETs.

An important matter to consider about latencies is the exact definition of the latency. The latency can be taken as the purely electrical or processing latency, but it can also include the mechanical latency of the (vibration) motors. In deliberation with SenseGlove, it was determined that latency would be defined as the time between the computer sending the data to the moment the system sends the signal to the actuators. So mechanical latency and latency within the PC software is not taken into account. Additionally, the latency of the microcontroller was not measured in the final design as it is already included in the latencies of the subsystems. The latency of the driver in the Palm vibrotactile Feedback department was hard to determine. This is due to the nature of the output, which is explained in their report [2]. Their latency was estimated based on the data sheets.

Table 3.2: Estimated latency budget.

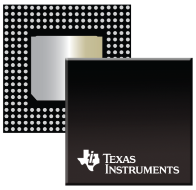
| Component | Estimated latency |
|---|---:|
| Wireless communication | 10 ms |
| Microcontroller | 1 ms |
| Per finger force feedback | 0.1 ms |
| Per finger vibrotactile feedback | 2.5 ms |
| Palm vibrotactile feedback | 4 ms |

Table 3.3: Measured latencies per subsystem.

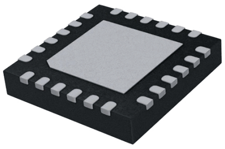
| Component | Measured latency |
|---|---:|
| Wireless communication | 7 ms |
| Per finger force feedback | 0.1 ms |
| Per finger vibrotactile feedback | 1.9 ms |
| Palm vibrotactile feedback | 0.1 ms |

## 3.5. Broad Design Choices

Some general design decisions were during the design process. Firstly, what component packages were going to be used. Since everything had to be soldered by hand BGA packages would be very difficult to solder properly. As can be seen in Fig. 3.1a the package has pins on the bottom which are very hard to reach during soldering. BGA is therefore avoided. The same goes for QFN packages, while they are easier to solder than BGA, they still pose a challenge, however, the QFN package ended up being almost impossible to avoid in some cases. In Fig. 3.1b the QFN package is shown, it can be seen that the soldering pads are on the bottom but also reachable from the side. Another component choice was regarding the size for the passive components like resistors, capacitors, etc. Of course having smaller components would lead to an overall more comfortable design for the glove. This is due to a better fit on the wrist, because of the smaller PCB size. However, this would again make it hard to solder by hand. Therefore the imperial 0805 component size was chosen as a good compromise between size and ability to solder by hand. However, for the final prototype the space constraints were so tight that for the Finger Vibrotactile feedback subsystem, components with the size of 0603 were chosen. Another decision with a major impact on form factor was the amount of layers of the PCB. With more layers, less space is required to route all the wires as well as the fact that it improves power distribution and shielding, due to the ability to add more power and ground planes. The downside of going from a 2 to a 4 layer PCB is monetary cost, with a 4 layer PCB being almost twice as expensive [9]. For the first PCB a 2 layer design was made and manufactured. Because of this experience and space constrains it is decided to use a 4 layer PCB for the final prototype.



(a) BGA package [14].



(b) QFN package [15].

Figure 3.1: BGA and QFN packages

## 3.6. General System Overview

In Section 2.3 all subsystems that are integrated in the SoftGlove are discussed. In Fig. 3.2 an overview of all connections between this subsystems is shown. The subsystems are abbreviated by FFF for per finger force feedback, FVF for per finger vibrotactile feedback and PVF for palm vibrotactile feedback. The blue lines represent the data lines between the modules, where the numbers show the amount of data lines. The red lines represent the power lines between the modules with the voltages shown on the lines. The USB block represents an USB micro input to charge the battery and connect to program the microcontroller which is shown as the ESP32 block. Furthermore, the power conversions block consists of a buck converter to create the required 5 V as well as a boost converter to generate the 24 V for the finger force feedback.

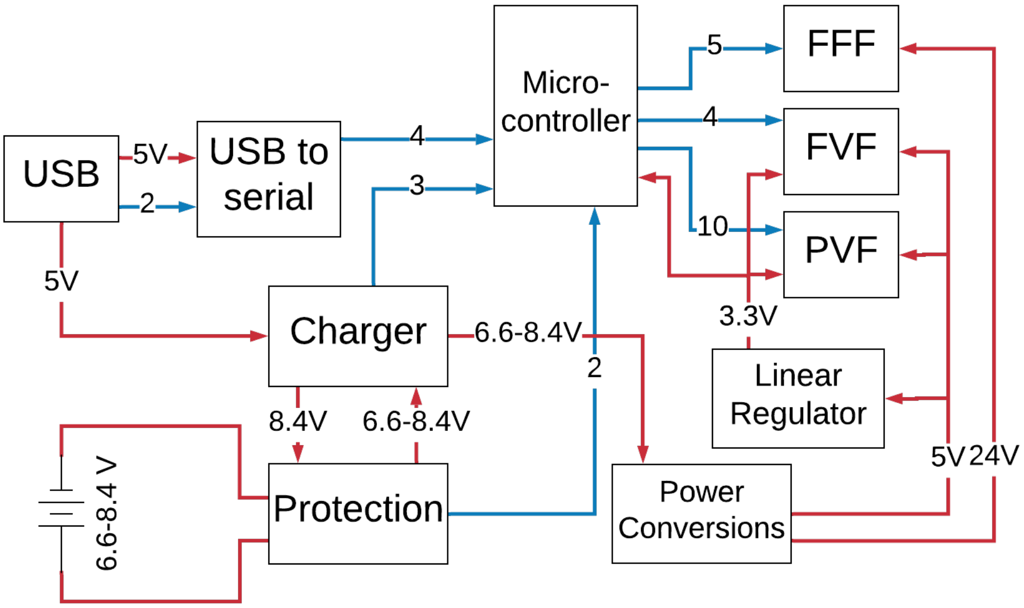


Figure 3.2: SoftGlove system overview. The subsystems on the top right are abbreviated as follows: Finger Force Feedback (FFF), Finger Vibrotactile Feedback (FVF) and Palm Vibrotactile Feedback (PVF).

All data lines are connected to the microcontroller. When determining all the data lines to the microcontroller, specifications had to be taken in account. First of all some pins output a PWM signal while the microcontroller is booting. Second, some pins are not allowed to be pulled up or down when the microcontroller is switching on. This is since these pins are responsible for selecting the boot mode. Third, some pins are specified to be just an input or just an output pin. The pin layout is therefore carefully designed and can be found in detail in Appendix A.7.

## 3.7. PCB Layout

The PCB stage consisted of two stages. A first PCB which is mainly focused on the functionality of the subsystems. The second PCB, which will be a revision of the first PCB, is mainly focused on the form factor and the placement of the subsystems. The second revisions will be the final prototype. The first PCB is 10.5 cm by 14.5

cm which is not the size that meets the requirement to fit on the wrist. The functionality of all subsystems is discussed and tested together with the revisions for the individual subsystem in the theses as described in Section 2.3. The layout of the second PCB, the final prototype, will be discussed in this section. As stated in Section 3.5, the first PCB is made with just 2 layers and the second PCB with 4 layers.

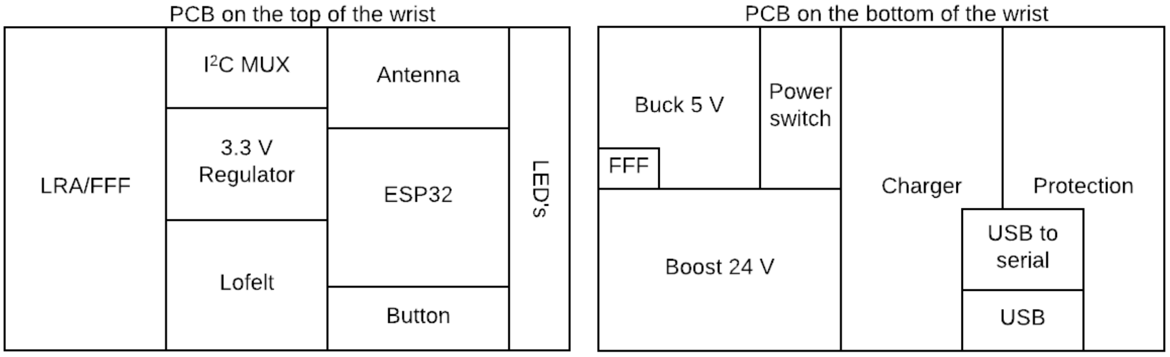
### 3.7.1. General Improvements for the Second PCB

After soldering and testing the first PCB, some general improvements had to be made when designing the second PCB. These improvements are listed below.

- A reset button for the microcontroller is needed.
- A power switch to turn the whole system on and off is needed.
- More test points need to be placed where possible.
- Pull-up resistors are required for both I$^2$C buses.
- Capacitors with a small capacitance need to be placed as close to the ICs as possible.

### 3.7.2. Final PCB Layout

All the improvements that are discussed in Section 3.7.1 together with the improved subsystems led to the final PCB layout that is shown in Fig. 3.3. The circuits schematics of the final PCB can be found in Appendix A.1. The final layout consists of two PCBs that both have a size of 40 mm by 70 mm, which is considerably smaller that the first PCB. The choice for two small PCBs gives the possibility to mount one PCB on the top of the wrist and the other one on the bottom of the wrist. Each PCB is mounted with one of the lithium-polymer cells, so a cell on the top and bottom of the wrist which can together deliver the 7.4 V. In Fig. 3.4 it is shown how this construction is set up. The PCB has all the components placed on one side to make sure nothing collides with the battery cells. The structure and design of all separate layers of the final complete PCB can be found in Appendix A.2.



(a) Top PCB layout.

(b) Bottom PCB layout.

Figure 3.3: The layout of all subsystems on the final PCB

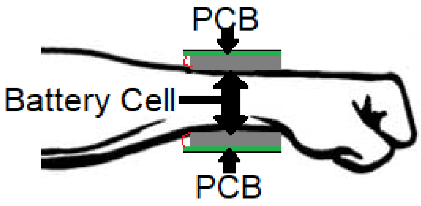


Figure 3.4: Mounting of the PCB and battery to the arm of the user.

# 4 Component Selection

In this chapter the reasoning behind the component choices is given. First the human perception of vibrations is discussed after which the type of actuator used is examined. This actuator needs to be driven, the way this is achieved is discussed last.

## 4.1. Human Perception of Vibrotactile Feedback

An important part of the development of haptic feedback devices is to capitalize on the difference between the feedback that is given by a device and the feedback that is perceived by the user. Human tactile sensitivity is not the same on all parts of the body, and some parts of the body are more often used than others. Additionally, humans cannot always accurately determine exactly where they are being touched [16], [17], especially when they are receiving several stimuli at the same time. In most applications, people either use the tips of their fingers separately, or touch things with their hand palm as well as their fingers. These two observations lead to the conclusion that, especially when a visual cue is present in order to trick the brain into expecting something [18], well chosen and placed actuators can be used to give a user the sensation of a wide range of interactions, with relatively few actuators.

In order to achieve this effect, it is important to choose actuators that cannot easily be localized by the user. This can be used to make the user feel like they touched something with their fingertip, when in reality the actuator is placed somewhere further down the finger, for example. The most sensitive place to place the vibrational actuators would be the fingertip [19]. However, as the finger force feedback needs to be placed there, there is no room for the vibrotactile feedback. Placing the vibrotactile actuators on the inside of the hand is also not a viable solution, as there is no room for actuators if the hand is closed. The actuators cannot be placed on the back of the distal phalanges either, as this is where the fingernails are, which are less sensitive to vibrations [19]. It is therefore decided that the finger vibrotactile feedback is placed on the intermediate phalanges of the fingers. This is not possible for the thumb, as the thumb does not have an intermediate phalanx. Therefore, the actuator will be placed on the proximal phalanx of the thumb.

Additionally, the actuators need to vibrate in a frequency range where the human skin is especially sensitive to vibrations, which is 200-450 Hz [19]. The detection threshold of a stimulus on the hand is 0.06 N [20]. This detection threshold does not differ significantly between male and female audiences and does not need to be adjusted for the difference between the dominant and non-dominant hand [21]. This is in line with cost factor 6 as discussed in Section 2.2.

## 4.2. Actuator selection

For small vibration motors, the main possible actuators are the Eccentric Rotating Mass (ERM) and the Linear Resonance Actuator (LRA). Other larger actuators such as the Lofelt (which is placed on the hand palm) and the C-2 tactor [22] are also available and discussed.

An ERM consists of an eccentric mass attached to a rapidly rotating axis, an example of a coin-type ERM can be seen in Fig. 4.1. The acceleration of the mass in a circular motion creates a vibration. Both the frequency and the vibration intensity of the motor are linked to the angular velocity of the mass. As the technology is not very complex, the actuators can be driven relatively easily through pulse width modulation (PWM) signals and furthermore many dedicated driver ICs are available at a low price. Downsides to ERMs are the fact that they have a relatively high start-up time and people are more easily able to identify their exact location on the hand than with LRAs [23]. Due to their high start-up time they cannot be used to create complex vibrations that feature a range of vibration strengths, but are mainly limited to pulsing [6]. Finally, due to size constraints, the only ERMs that can be used in the SoftGlove are coin type ERMs, which would vibrate in the plane tangential to the finger. This can not be felt as easily as vibrations perpendicular to the finger.

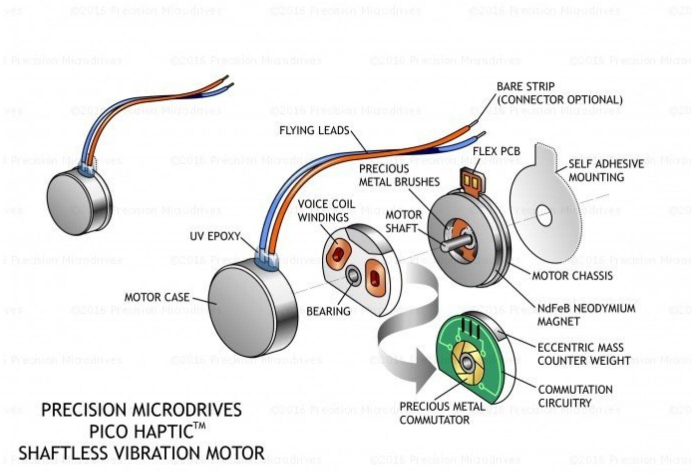


Figure 4.1: Representation of the internals of a coin-type ERM, taken from [24].

LRAs are made up of a mass attached to a magnet and a spring, an example of a coin-type LRA can be seen in Fig. 4.2. The magnet is moved using a voice coil. For the maximum vibration amplitude at a given voltage, this should occur at the resonance frequency of the spring, which the LRA is named after. The LRA can therefore only be driven at different frequencies at high cost in terms of vibration strength. On the other hand, the vibration strength can be controlled independently from the vibration frequency, unlike the ERM where both are related directly. This means that changing the vibration strength will not cause the vibration to shift to frequencies at which the hand is less sensitive to vibrations. Additionally, as the LRA needs to stay exactly at its resonance frequency to ensure proper operation, a dedicated driver IC is strongly recommended. Differences in vibration strength of LRAs can also more easily be detected by users than those of ERMs [16], leading to a richer VR experience. A nice feature for VR experiences would be to simulate the feel of different materials as described in [25]. However, due to the limiting frequency response this is not feasible to implement using a commodity LRA.

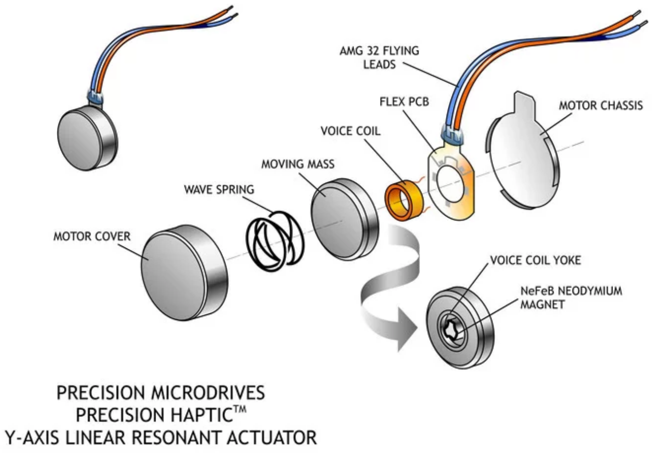


Figure 4.2: Representation of the internals of a coin-type LRA, taken from [26].

Some special types of LRA exist like the C-2 tactor or the Lofelt actuators. These allow for higher vibration

Table 4.1: Characteristics of the chosen LRA and current ERM as provided by the manufacturer, [27], [28].

| Name | Diameter | $V_{\text{rated}}$ | $V_{\text{operating}}$ | $I_{\text{max,rated}}$ | $I_{\text{typical}}$ | $f_{\text{resonance}}$ | vibr. strength |
|---|---|---|---|---|---|---|---|
| G0832022D | 8 mm | 2.0 $V_{RMS}$ | 0.1-1.8 $V_{RMS}$ | 90 mA | 68 mA | 235 Hz | 1.50 G |
| C1020B217F | 10 mm | 3 $V_{DC}$ | 2.7-3.3 $V_{DC}$ | 90 mA | 61 mA | N/A | 1.10 G |

intensities at a larger frequency range. The downside to these however is that they are proprietary making them relatively hard to obtain and more expensive. Aside from this they are also quite large when compared to a regular LRA, with a diameter of about 3 cm, making them unsuitable for placement on the fingers.
As the latency of LRAs is smaller than that of ERMs and more complex haptic effects are possible using LRAs, LRA motors were chosen to be implemented in the SoftGlove. This leads to an additional need for LRA drivers, as discussed in Section 4.3.

### 4.2.1. LRA selection

The four metrics that were considered when choosing an LRA were size, shape (coin type), vibration direction (perpendicular to the finger) and vibration strength. The LRAs used for the finger vibrotactile feedback need to have a strength at least equal to the ERMs that are currently in the SenseGlove. The current ERM vibration motor, C1020B217F by Jinlong Machinery [6], has a vibration strength of 1.1G. G is the conventional unit for vibration strength used by manufacturers. It is meant to describe the gravitational acceleration at sea level (9.81 m/s$^2$) for a test mass of 100 g. 1 G therefore approximately corresponds to a force of 1N. As discussed in Section 4.1, the threshold for detection on the human hand is 0.06 N. Therefore, the human hand can detect vibrations starting from about 0.06G, which means that the maximum vibration amplitude of the implemented LRA should be (much) larger. The LRA chosen was G0832022D by Jinlong Machinery, [27]. The most important specifications of this LRA are given in Tab. 4.1.

## 4.3. Haptic driver selection

Due to the high Q factor of LRAs [29] it is essential that they are driven at the resonance frequency [30], [31]. If this is not done the vibration would be unsatisfactory due to the limited vibration strength. Additionally, due to normal manufacturing tolerances some LRAs actual resonance frequency would be closer to the driving frequency than others, resulting in too much variance between the vibrations at different fingers. To make sure that the LRAs are driven at their respective resonance frequencies, driver chips can be used. The driver chips available were mainly made by Texas Instruments as well as from ON Semiconductor, Dialog Semiconductor and Diodes Incorporated. For the prototyping stage it is inconvenient to use ball grid array (BGA) packages since they are very difficult to solder by hand. Apart from this some features were sought for in these drivers, namely automatic resonance tracking (as explained above), automatic braking and overdrive as well as I$^2$C control. This means that the driver applies a larger voltage than necessary for the desired effect for a short time to arrive at the desired vibration amplitude more quickly. For example, when the LRA should go from being completely still to 50% of its maximum vibration strength, it might be driven in the same way as it would be to make it vibrate at 100% until it is vibrating at 50%, after which the driver goes back to driving it to 50% of its maximum vibration strength. This makes the rise time shorter, braking works similarly for the fall time. Finally, the I$^2$C control was chosen as it allows several LRA drivers to be accessed from the same pins on the microcontroller. This means that relatively few pins will be needed for the finger vibrotactile feedback, which allows more pins to be used by the other subgroups and for testing signals. The driver chips that remained were the Texas Instruments DRV2604L, the Texas Instruments DRV2605L and the Dialog Semiconductor DA7281.

A benefit of the DA7281 is the selectable I$^2$C address. The chip allows for 4 different I$^2$C addresses to be set using two pins. Mandatory requirement 2 specifies that vibrotactile feedback is desired on each finger. Therefore 5 LRAs and thus 5 drivers are needed. Since almost no interactions occur using just the pinky or the ring finger individually [20], one might suggest to drive 2 LRAs with one driver for these fingers or have 2 drivers with the same address such that they both drive the attached LRAs in the same way. These driver chips use measurements of the back electromotive force (BEMF) to determine and track the resonant frequency of the LRA. Since the driver expects only one LRA to be attached this measurement is thrown off when attaching multiple LRAs to one driver and could produce erroneous results and is therefore not recommended by the manufacturer [32]. As for the other option of having two devices with the same I$^2$C address on one bus reading

from the drivers will go wrong. The SDA line in the I$^2$C protocol is normally pulled up, devices will pull down the SDA line in order to transmit a 0. Reading data from two devices with the same address will therefore only result in a read 1 if both devices try to send a 1, effectively resulting in the AND operation on the register values of the devices. The main advantage of the DA7281 is thus not applicable to this use case.
Since the DRV2605L is available on a breakout board by Adafruit it was a convenient choice to use in the proof of concept stage of the project. Furthermore, since the DRV2604L has the exact same pin layout, after testing the DRV2605L could be desoldered from the breakout and could be replaced with a DRV2604L. This makes the chips easy to test in the proof of concept stage. Therefore the DRV2604L and the DRV2605L were chosen to test in the proof of concept stage rather than the DA7281.

### 4.3.1. DRV2604L and DRV2605L comparison

The information about the DRV2604L and the DRV2605L are taken from their data sheets, [33] and [34] respectively. The drivers share their main features, such as the ERM and LRA compatibility, automatic resonance tracking, automatic overdrive and braking, I$^2$C mode drive and more. The differences between the two mainly consist of the use of the internal RAM. On the DRV2605L the RAM is preloaded with the TouchSense 2200 library. This is a proprietary library of 123 haptic effects including different clicks, hums and buzzes at different intensities. The contents of this library can be seen in Appendix B. The DRV2605L additionally has the option of converting an audio signal to vibrations. Since the bandwidth for data and the latency are limited this would not be a feature that would be used in the SoftGlove. The DRV2604L on the other hand exposes its internal RAM to the user, who can therefore load in wave forms themselves and trigger them in the same way as the TouchSense 2200 library on the DRV2605L. Unfortunately this RAM is not static, meaning that it has to be reprogrammed every time power is cycled. It can, however, be used to load in the required haptic wave forms for a particular scene of the game or simulation during a loading screen, which can then be triggered to cut down on the data that needs to be transferred during operation. Since the DRV2605L includes the licensing costs of the TouchSense 2200 library, the cost of the component is also higher, €0.96 according to Digi-Key (retrieved 14-6-2019). Due to the fact that these effects are preprogrammed they offer very little flexibility, which is desired since the SenseGlove can be used in a multitude of applications. The DRV2604L would therefore be preferred since the extra functionality of the DRV2605L is either not useful or limiting while the extra functionality of the DRV2604 actually increases flexibility.

### 4.3.2. Modes of operation

Using the DRV2604L the LRA can be driven in several different ways. Namely by using the integrated RAM for storing wave forms, using the Real-Time Playback (RTP) mode, by making use of the PWM input or by using the analog input. The following sections will explain how these modes operate.

**wave form memory** The DRV2604L has 2 kB of integrated RAM that can be used to store user made wave forms. These wave forms are programmed in by the user and can be queued up an subsequently played back. The memory can contain up to 127 different wave forms. This RAM is not static and therefore has to be reprogrammed every time the system is turned on.
The wave forms are specified as voltage time pairs with 7 bits available to specify the voltage and 8 bits to specify the time. The time value is specified as a number of ticks, by default a tick occurs every 5 ms. The time and voltage values are placed in two 8 bit registers. This leaves 1 bit of space which is used to specify whether linear interpolation needs to occur between the time and voltage pairs.
wave forms are to be programmed in by writing the voltage time pairs to a specific set of registers over I$^2$C. Apart from this a header block needs to be written that specifies which registers contain the wave form data, specifically the start addresses of the upper and lower byte. The header block also contains another byte that specifies the effect duration and the number of repeats. If a specific wave form needs to be repeated 3 bits can be used to specify this amount. If this value is set to 7 the effect will be repeated indefinitely until the GO bit is cleared. The other 5 bits are used for the size of the effect, specifically the amount of time and voltage bytes the effect consists of. This value is therefore restricted to even numbers between 2 and 30.
When these wave forms are programmed, they can be triggered in a number of ways. Firstly the wave forms need to be placed in the wave form sequencer. The wave form sequencer consists of 8 bytes that need to be written with the ID of the desired wave form over I$^2$C. After the desired wave forms are queued up they can be triggered depending on the mode by writing the GO bit over I$^2$C or by making use of the IN/TRIG pin.

**RTP mode** In this mode a register can be written over $I^2C$ with a desired amplitude value. The driver will use its automatic resonance tracking to drive the LRA at the desired amplitude at its resonance frequency.

**PWM input mode** In this mode the DRV2604L accepts a PWM signal between 10 and 250 kHz. The duty-cycle of this signal is then linked directly to the vibration amplitude, with 50% duty-cycle resulting in half of the maximum vibration amplitude.

**Analog input mode** This mode functions practically the same as the PWM input mode however with an analog voltage instead of a PWM signal. The maximum input voltage of 1.8 V is linked to the maximum vibration amplitude, 0.9 V to half of the maximum vibration amplitude and 0 V to rest.

The chosen mode of operation is the RTP mode. This has the benefit of being compatible with the existing SenseGlove Communication Protocol [11], which is a stretch goal as can be seen in Section 2.2. Furthermore, it allows the driver to be used while communicating solely over $I^2C$ which is desirable due to the limited number GPIO pins available on the microcontroller.

### 4.3.3. Data format

While driving an LRA the DRV2604L can operate in two different modes, namely open-loop and closed-loop mode. In open-loop mode the automatic resonance tracking is disabled and the LRA is driven at a frequency specified by the user. This allows the LRA to be driven outside of its resonance frequency, this can be useful when using an LRA with a wide frequency response like the Lofelt actuator. Apart from not tracking the resonance frequency the driver also does not apply automatic overdrive and braking. In some cases having this feature disabled might be preferred since the overdrive and braking can already be built in to the wave form that is sent to the driver. However this is not the case for the SoftGlove and therefore closed-loop mode is preferred, because of the automatic resonance tracking.
Furthermore, there is the distinction between unidirectional and bidirectional input for closed-loop mode. Where setting the device to bidirectional mode makes it compatible with wave forms created for open-loop operation while still being able to make use of the features of closed-loop mode. In this mode half of the input range is dedicated to braking, therefore allowing for less detailed steps in the vibration amplitude. Again, since no compatibility with existing braking is required the unidirectional input mode was chosen for operation as this allows more control over the vibration strength level.

### 4.3.4. Auto calibration

The DRV2604L has an auto calibration feature to set specific parameters that the driver uses to drive the LRA optimally. Variations in the characteristics of the LRA occur even when using the same model. The parameters that are determined by this calibration are the analog gain of the BEMF amplifier, compensation for resistive losses in the driver and the BEMF voltage the actuator gives when driven at the rated voltage.

The instructions for performing the auto calibration [33] provided by the manufacturer were followed. However, the auto calibration was never completed successfully. On the support page of the manufacturer switching to the newer DRV2625 or DRV2624 drivers was recommended, as this enables the user to read the diagnostic data as to why the auto calibration fails [35]. The currently used DRV2604L driver only reports either a failed or a successful calibration. These drivers are, however, only available in a ball grid array (BGA) package. It was decided that no BGA packages would be used in the prototype, the reasoning for which can be found in Section 3.5. Because of this the auto calibration feature was not used for the prototype.

# 5
# System Integration

The finger vibrotactile subsystem is not intended for standalone use, and therefore has to be integrated with the whole system. The design choices that resulted from this are discussed in this chapter.

## 5.1. Controlling multiple I$^2$C slaves with the same address

Since each finger will get its own LRA and thus its own driver, five drivers will be present on the SoftGlove. The selected driver chip unfortunately does not support changing its I$^2$C address. This causes a conflict since all driver chips will be on the same I$^2$C bus. This is not an uncommon problem and thus there are various ways that this conflict can be resolved.

### 5.1.1. I$^2$C multiplexer and I$^2$C switch

An I$^2$C multiplexer is a device that connects the input I$^2$C channel to one of its output channels. These devices often have their own I$^2$C address, with a register that can be written to select which channel is active. An I$^2$C switch is very similar to an I$^2$C multiplexer. However, it has the added ability to output on multiple channels at once. The TCA9548A I$^2$C switch by Texas Instruments provides 8 output channels, while also being able to act as a level shifter. This switch comes in a TSSOP-24 package and costs €1.39 on Digi-Key (retrieved 13-6-2019). The latency characterization of this component can be seen in Chapter 6.1.

### 5.1.2. I$^2$C address translator

An I$^2$C address translator is a device that connects to the SCL and SDA lines between the master and the slave. It uses resistors connected to two input pins to specify an 8 bit value which is then bitwise XORed with the address byte when it is recognized on the I$^2$C bus. This makes for a low latency solution, with a specified typical delay of 170 ns between the in and output SDA channel [36]. In this case 4 address translators would be required to have a unique address for all 5 LRA drivers, as one driver could keep te original address. This solution would be fairly expensive in terms of both space on the PCB and money, as 4 components would need to be added, with a single LTC4316 address translator by Analog Devices costing €3.62 on Digi-Key (retrieved 13-6-2019) while coming in an MSOP-10 package.

### 5.1.3. Comparison

An address translator would provide a lower latency solution over an I$^2$C switch or multiplexer. However, since 4 address translators would be required to give each LRA driver a unique address this would be around 10 times more expensive, as well as taking up quite a bit more space on the PCB. Since the size of the PCB is a big cost factor and the difference in latency is quite minor this solution is less favorable than using a switch or multiplexer. Finally the TCA9548A I$^2$C multiplexer was chosen for having at least 5 output channels, voltage level shifting and availability in a TSSOP package. The impacts that this multiplexer has on the latency of the system can be seen in Chapter 6.1.

## 5.2. Operating voltages

All ICs of the finger vibrotactile feedback subsystem can run on a range of voltages. The I$^2$C switch can operate on 1.65 V to 5.5 V while the LRA driver can operate on 2 V to 5.2V . These supply voltages do have certain restrictions, both for the switch and the LRA driver. The LRA driver can not drive the LRA at a higher voltage than the supply voltage. Since the LRA has a rated voltage of 2 V RMS both 3.3 V and 5 V for the LRA driver would be sufficient.

The hand palm vibrotactile feedback subgroup requires a 5 V rail for their audio amplifier and since the LRA drivers can also function on this voltage, it was decided to step down from the battery voltage to 5 V with a buck converter. This 5 V is stepped down further to 3.3 V for the microcontroller with a low dropout regulator. This is not an efficient conversion, but due to the small difference in voltage and the small power draw the effect on the power draw of the system is minor. The buck converter, however, can be fairly efficient and it was therefore decided to have the LRA drivers operate at 5 V. The power consumption was tested with a driver running at 5 V, driving at maximum amplitude. The peak current used by the driver and the LRA was around 60 mA. This means that the whole finger vibrotactile subsystem has a peak power consumption of about 1.5 W.

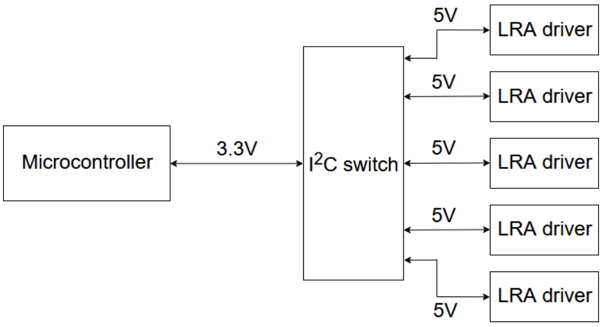


Figure 5.1: Schematic overview of the I$^2$C buses and their operating voltages.

Since the LRA drivers have to operate on 5 V, they expect an I$^2$C bus running on 5 V. However the microcontroller runs on 3.3 V and therefore runs its I$^2$C bus on 3.3 V as well. A schematic illustration of this can be seen in Fig. 5.1. Level shifting needs to take place, this is built in to the used I$^2$C switch. In order to make sure that the switch is able to effectively clamp the bus voltages, meaning that it can keep the voltage levels separated, its pass voltage needs to be less than or equal to the lowest bus voltage. In this case the lowest bus voltage is the 3.3 V bus between the microcontroller and the switch. This pass voltage is related to the supply voltage of the switch, the exact relation can be found in the data sheet [37]. In order to keep the pass voltage below 3.3 V the supply voltage has to be lower than about 4.6 V, therefore the switch is driven at 3.3 V.

# 6 System Characterization

The data sheets of the used components give an indication to the performance of the individual components. However, in the final product they are integrated and may therefore have different characteristics. In order to investigate these characteristics, tests were performed to characterize the system in terms of latency and force.

## 6.1. Latency Characterization

Latency tests were performed during the proof of concept stage as well as on the first PCB. Testing using the prototype was done to ensure that the characteristics of the system still matched the expectations from the proof of concept stage. A number of tests were performed to find the latencies of the different actions and components of the system.

### 6.1.1. Test setup

For the testing in the proof of concept stage a WEMOS D1 mini was used, which is an ESP8266 based microcontroller, using an $I^2C$ bus running at 100 kHz. Testing of the final prototype was done using the integrated ESP32 microcontroller with an $I^2C$ bus running at 400 kHz. A Tektronix TDS2022C oscilloscope was hooked up to the output of the LRA driver, in parallel with an LRA, while the external trigger input of the oscilloscope was connected to a GPIO pin of the microcontroller. This pin is used to signify the start of the test, after which the $I^2C$ command is sent to the LRA driver. Since the output of the LRA driver is hooked up to the oscilloscope, the exact moment at which the driver starts to output can be measured. By utilizing a GPIO pin as a trigger the start time of the measurement can be determined as well. The oscilloscope was set up to capture a single shot signal to make sure the oscilloscope triggers only once when the GPIO pin is made high, an example is shown in Fig. 6.1.

### 6.1.2. Results

The tests were performed both with and without the use of the $I^2C$ switch. This was done to be able to separate the latency of the LRA driver itself and the added latency of the switch. Additionally, testing was performed with the switch between the microcontroller with and without changing the output channel.

The theoretical minimum latency of this setup can easily be calculated. Since the $I^2C$ protocol sends only 1 bit per clock cycle the equation becomes $t_{\text{transfer}} = \frac{\#\text{bits}}{f}$. For the direct tests, the LRA driver is directly attached to the microcontroller. For this the data to be transferred consists of a start bit, the driver's $I^2C$ address byte, the byte specifying which register to write to, the data byte, the stop bit and an acknowledge bit sent by the slave device after each byte. This transaction consists of 29 bits which results in a transfer time of 0.29 ms with an $I^2C$ bus running at 100 kHz and 0.0725 ms while running at 400 kHz. For the tests including a channel change the amount of bits transferred increases, before the microcontroller can communicate with the driver it first needs to set the correct channel. The setting of the channel takes 20 bits consisting of a start bit, the switch's $I^2C$ address, a data byte, a stop bit and two acknowledge bits. This means that 49 bits need to be sent, resulting in a theoretical latency of 0.49 ms at 100 kHz and 0.1225 ms at 400 kHz.

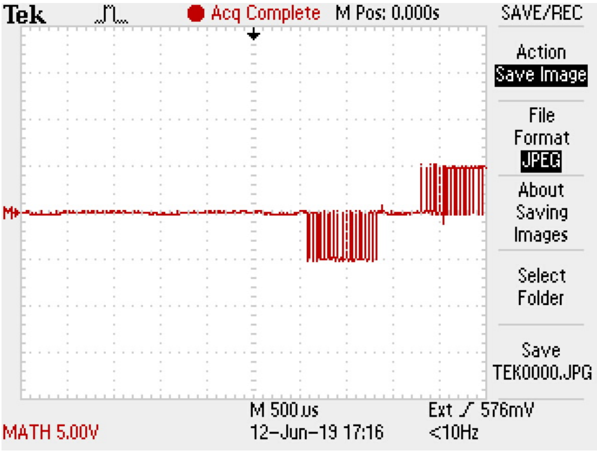


Figure 6.1: An example of the oscilloscope output during a latency measurement. The arrow signifies the point where the oscilloscope was triggered by the GPIO pin.

Of course the driver and the switch also have some internal delays, resulting in overall higher latencies. A summary of the results can be found in Tab. 6.1 while the graphs for all measurements can be found in Appendix C. The arrival rate of the driving signal of the oscilloscope can be modelled as a Poisson arrival process. As each measurement is independent and follows an identical Poisson distribution, according to the central limit theorem the distribution will approach a Gaussian continuous distribution if enough measurements are taken [38]. Therefore, all tests were performed 50 times in order to have a reasonable approach to a Gaussian distribution. More measurements would result in a better fit, however, due to time constraints this value was chosen.

The most important measurements are the ones that include a channel switch since that will be used in the final system. This latency was first measured using the Arduino framework based on C with the different components on breakout boards. The exact same measurement was repeated on the first prototype PCB and using the ESP-IDF framework. Even though the final prototype has an I$^2$C bus running at 400 kHz these tests were performed with the I$^2$C bus running at 100 kHz. This was done because the microcontroller used to perform the initial tests supports a maximum frequency of 100 kHz [39]. Results of these measurements can be found in Fig. 6.2 and 6.3 for the Arduino and ESP-IDF frameworks respectively. On average the latency using the ESP-IDF framework is lower. This is to be expected since the ESP-IDF framework provides lower level access which therefore results in better performance. A summary of the measured latencies using the ESP-IDF framework can be found in Tab. 6.2.

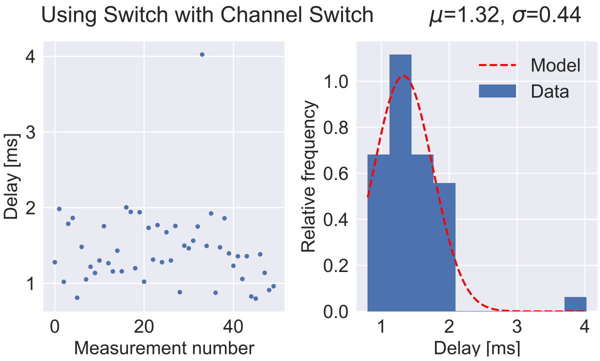


Figure 6.2: Latency measurement with channel switch on breadboard using the Arduino framework, with an I$^2$C bus running at 100 kHz.

Since the microcontroller on the prototypes supports a maximum I$^2$C bus speed of 5 MHz [40], latency tests were also performed at speeds higher than 100 kHz. Even though the driver and the switch have a specified maximum speed of 400 kHz they worked at a bus speed of 1 MHz. Going above this speed, however, caused the I$^2$C transactions to fail. The results of the latency test performed at 1 MHz can be found in Fig. 6.4. The

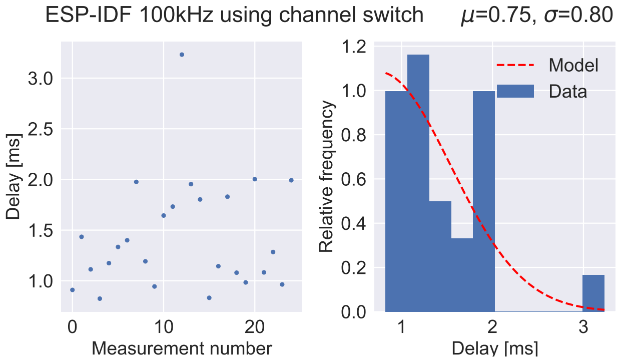


Figure 6.3: Latency measurement with channel switch on the first prototype PCB using the ESP-IDF framework, with an $I^2C$ bus running at 100 kHz.

mean value of this test is actually 0.02 ms higher than the test performed at 100 kHz. However, considering the relatively high standard deviation (0.8 ms), this could also be due to a deviation between the sample mean and actual mean as a result of the low number of measurements. As the difference is sufficiently small to be insignificant and given the limited amount of time the authors had at their disposal, the authors chose not to perform more measurements. The fact that the latency does not improve can have multiple reasons. Firstly $I^2C$ devices can perform clock stretching. Normally the SCL line is controlled by the master, the microcontroller in this case, however when a slave device is not able to keep up, it can hold down the clock to reduce the bus speed. Secondly, the $I^2C$ communication could not be the limiting factor in the latency. Meaning that the latency is dominated by the internal delays of the driver and switch. The clock stretching is most likely not the main limiter since both the switch and the driver both support a maximum frequency of 400 kHz, which would be an improvement over 100 kHz. Therefore, the time taken by the $I^2C$ communication is investigated further in Section 6.1.3.

Some outliers can be seen in for example Fig. 6.2 and 6.3. This can have a number of reasons but it is hard to say for certain what this caused. The microcontroller could have gotten an interrupt in the time that it is supposed to send the data to the driver, thereby introducing an additional delay. The LRA drivers presumably use a kind of microcontroller internally, when the amplitude is written right after a clock cycle an extra latency is introduced.

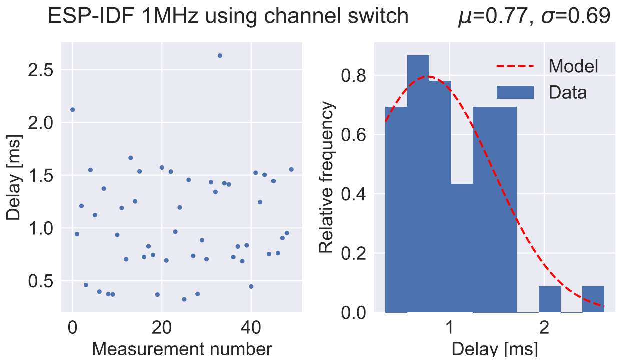


Figure 6.4: Latency measurement with channel switch on the first prototype PCB using the ESP-IDF framework, with an $I^2C$ bus running at 1 MHz.

Table 6.1: Summary of the tested latencies using the Arduino framework.

| Type of test | Latency [ms] |
|---|---|
| Direct, RTP mode | 1.0 ± 0.5 |
| Direct, library, $I^2C$ trigger | 1.19 ± 0.02 |
| Direct, library, trigger pin | 0.95 ± 0.03 |
| Switch, RTP mode, no channel change | 0.64 ± 0.97 [1] |
| Switch, RTP mode, with channel change | 1.3 ± 0.4 |

Table 6.2: Summary of the tested latencies on the first prototype PCB using the ESP-IDF framework.

| Type of test | Latency [ms] |
|---|---|
| RTP mode, 100 kHz $I^2C$ bus | 0.8 ± 0.8 |
| RTP mode, 1 MHz $I^2C$ bus | 0.8 ± 0.7 |

### 6.1.3. Execution time

The latency of the system includes both the internal delays of the LRA drivers as well as the $I^2C$ switch. The microcontroller only has to perform the $I^2C$ communication with the drivers until it can move on to drive the other components like the Lofelt or the finger force feedback. The time taken by the microcontroller to send an amplitude value to each channel in succession was measured by using the ESP32's internal timer. This timer has a microsecond resolution. The resulting delay was 1.433 ms to set an amplitude value for each of the five channels individually. This testing was performed with an $I^2C$ bus speed of 1 MHz.

## 6.2. LRA Characterization

In order to characterize the LRAs, four characteristics were determined, namely: the resonance frequency, the vibration amplitude, the rise time and the fall time.

### 6.2.1. Test setup

To perform these measurements an accelerometer is required. In order to satisfy the Nyquist criteria the samples needed to be taken with a frequency of at least twice the resonance frequency. Since the specified resonance frequency of the chosen LRA is 235 Hz, this results in a sampling rate of at least 470 Hz. Furthermore the LRA was specified for a vibration force of 1.5 G, where G is the gravitational acceleration in $m/s^2$, meaning that the accelerometer needs to have a range of at least 1.5 G. Because of these specifications the ADXL345 accelerometer by Analog Devices [41] was chosen. This accelerometer features a maximum sampling rate of 3200 Hz and a 10 bit resolution in the ±2 G range along with support for both $I^2C$ and SPI communication. Unfortunately the maximum output data rate supported for 100 kHz $I^2C$ is 200 Hz [41] which is not enough to satisfy the Nyquist sampling criterion. Therefore the SPI protocol needs to be used to communicate with the accelerometer. Since the microcontroller supports a maximum SPI frequency of 80 MHz [39] and the accelerometer 5 MHz, the SPI bus was used at 5 MHz. This frequency provides enough bandwidth to run the accelerometer at its maximum output data rate.

The accelerometer was mounted on a 100 g weight with the LRA attached on top of that using its included double sided tape. The weight was then placed on top of a piece of foam to allow the system to vibrate without much resistance and to isolate it from vibrations in the surface on which the measurement was conducted. A schematic of the test setup can be seen in Fig. 6.5. This measurement setup was chosen in order to follow the LRA manufacturer's test setup as closely as possible, this setup can be found in Appendix D.1. With the available materials it was impossible to mount the LRA and accelerometer in such a way that they were perfectly aligned with one axis. Therefore, a combination of all 3 axes needed to be considered during data reduction. This is elaborated on in Section 6.2.2.

[1] Measurement error too large to provide accurate results. Performing more measurements would likely provide a more precise result. As the switch will always perform a channel change between activating drivers in the final product, the group, however, decided not to prioritize this.

The interrupt pin of the accelerometer is set up to go high when new data is available, when this is detected the accelerations in the x, y and z directions are read from their six registers. These six registers had to be read in a single SPI transaction to make sure that the data remained constant during the read-out process. Since the microcontroller does not feature a large amount of memory this data needed to be offloaded. This was done by transmitting the data over a serial connection to a PC. Each sent line consists of the acceleration in the x direction then in the y direction and then in the z direction with all values being separated with a comma, which could later be loaded in for processing as a csv-file. In order to send the data quickly enough the serial communication was run at 1 MHz. This data was then received and logged using the program PuTTY, which is an open source terminal emulator. The captured data was then further processed using Python to generate plots and analyze the data.

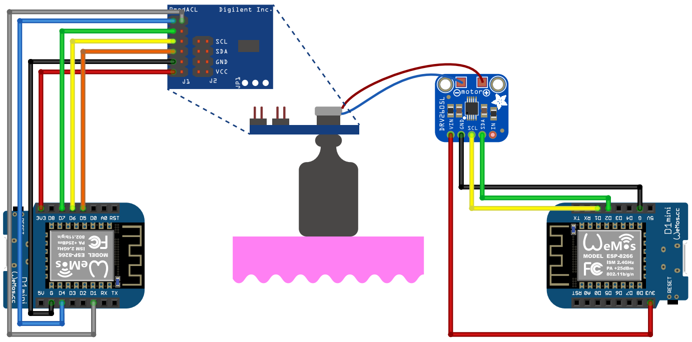


Figure 6.5: Test setup used for force characterization. On the left, the microcontroller that reads out the accelerometer is shown. It is connected to the accelerometer break-out, which is shown both in top view and side view. In the side view representation, the LRA is mounted on top of it, whereas the 100g mass and pink foam are shown below the accelerometer. The LRA is connected to the DRV2604L driver on the right. Finally, at the rightmost the microcontroller that controls the LRA driver is shown.

### 6.2.2. Data Reduction

This subsection discusses how the different characteristics of the LRA are found using Python. The data consists of measurements spanning several seconds in which the LRA is at rest at first, then accelerated to its maximum vibration strength, vibrates at maximum strength for a while and then is braked back to standstill again.

**Calibration** First, the accelerometer needed to be calibrated. This is done by orienting the accelerometer such that two axes are aligned tangentially to the ground and one axis is aligned perpendicularly to the ground and then recording the acceleration the accelerometer measures. This experiment is repeated such that each axis of the accelerometer has been oriented perpendicular to the ground. The axes tangential to the ground should not observe an acceleration, whereas the perpendicular axis measures an acceleration of 1 G, which is due to gravity. In this way, the gain of the accelerometer could be determined by again fitting a Gaussian to the measured output bits. The gains can be found in appendix Tab. D.1.

For each subsequent measurement, the equilibrium of each axis was calculated by taking the mean of 2000 measurement points from before the LRA started moving. This mean was subsequently subtracted from the entire data series in order to bring the equilibrium position to 0. Then, the gain was used to convert the acceleration to $\mathrm{m/s^2}$.

**Vibration Strength** Since each axis vibrates in a sinusoidal manner, the amplitude of the vibration can be calculated using the RMS deviation from equilibrium. In order to characterize the vibration amplitude, only the part of the measurement where the LRA was vibrating was used. In this area the RMS value of the acceleration was calculated and subsequently multiplied by $\sqrt{2}$ in order to obtain the amplitude, as described in Eq. 6.1. In this equation $a_{j,i}$ describes the acceleration of a data point $i$ on axis $j$ ($j = x, y, z$). After this is done, the norms of the amplitude vectors could be used to calculate the total vibration amplitude using Eq. 6.2.

$$A_j = \sqrt{2} \cdot a_{j,\mathrm{RMS}} = \sqrt{2\overline{a_{j,i}}^2} \tag{6.1}$$

$$A = \sqrt{\sum_{j=x,y,z} A_j^2} \tag{6.2}$$

**Resonance Frequency** The resonance frequency was again found using only the parts of the measurement where the LRA was vibrating. A sinoid was fitted to all three axes of the measurement. Since the baseline had been removed during calibration the equilibrium of the sinoid was known to be zero. Additionally, the amplitude was already known from Eq. 6.1. Therefore, each sine only had two degrees of freedom, namely the frequency (which is the resonance frequency of the LRA) and the phase offset. As the vibrations all describe a part of the same vibration, the frequency of all sinoids was taken to be equal, whereas the phase offsets were allowed to be off by $\pi$, as a phase shift of $\pi$ simply means that an axis is defined in the opposite direction with respect to the vibration. The sine was fitted using the modified least squares cost function in Eq. 6.3. In this function $a_{\text{model},\phi_0,j}(t)$ describes the modeled acceleration at time $t$ with phase offset $\phi_0$ rad in direction $j$. The frequency is described by $f$ and the cost function by $\Lambda$. $\Lambda$ was subsequently minimized using the `scipy` package in Python.

$$\begin{aligned} a_{\text{model},\phi_0,j}(t) &= A_j \sin(\phi_{0,j} + 2\pi f t) \\ \Lambda &= \sum_{j=x,y,z} \min\left\{ \sum_i^n [a_{j,i} - a_{\text{model},\phi_0,j}(t=t_i)]^2, \sum_i^n [a_{j,i} - a_{\text{model},\phi_0+\pi,j}(t=t_i)]^2 \right\} \end{aligned} \tag{6.3}$$

**Rise and fall time** In order to describe the rise and fall of the vibration, an activation-function-like envelope for the sine was used. The equation is shown in 6.4. In this equation, $t_0$ describes where the envelope passes 50% of $A_j$ and $\alpha$ is a variable that describes the horizontal stretch of the envelope. The rise and fall times could subsequently be calculated using 6.5. The rise and fall times were defined as the time the envelope takes to go from 10% of $A_j$ to 90% of $A_j$. This is different from the way in which the manufacturer defines the rise and fall times [27], as they define it as the time until the vibration reaches 50% of $A_j$. The manufacturer used the time at which the driver started outputting a signal as the start of this timespan. With the available materials, the authors were unfortunately unable to measure the exact moment when this happened in the reproduced setup. The 10% to 90% was therefore adapted by calculating when the envelope reached 10 % of its final amplitude and when it reached 90 % of its final amplitude.
Unfortunately, due to a slight instabilities in the area around the activation and deactivation time `scipy` was unable to find a good fit to the data using the multiplication of the sine and envelope. Several fitting methods, including Markov chain Monte Carlo (MCMC) were attempted, however, none of the fits produced usable results. This is most likely due to the limited amount of data points in the activation/deactivation region. Due to time constraints, the group therefore resolved to make the envelope fits by eye.

$$a_{\text{envelope},j}(t) = \frac{A_j}{1 + e^{-\frac{(t-t_0)}{\alpha}}} \tag{6.4}$$

$$t_{\text{rise,fall}} = t_{90} - t_{10} = -\alpha \left[ \log\left(\frac{1}{0.9} - 1\right) - \log\left(\frac{1}{0.1} - 1\right) \right] = -\alpha \left[ \log\left(\frac{1}{81}\right) \right] \tag{6.5}$$

### 6.2.3. Results

All results of the data reduction described above can be found in Tab. 6.3. It is clearly visible that the measured amplitude at full power (0.72G) is much lower than that listed in the data sheet (1.50G, as can be found in Tab. 4.1). The manufacturer is unfortunately not very clear on their definition of the vibration strength, so it is possible that the listed value is a peak-to-peak acceleration rather than the amplitude. In that case the amplitude given by the manufacturer could more accurately be compared to the peak-to-peak acceleration, resulting in a value of 1.44 G, which is much closer. The difference could also be due to differences in the measurement setup.

The amplitude is also shown as a function of the set amplitude in Fig. 6.6. As the delivered amplitudes do not linearly scale with the requested amplitudes, more amplitude measurements were added to investigate this effect, especially in the region between 37.5 % (0x60 on the driver) and 50 % (0x7F on the driver). The resulting amplitudes can approximately be divided into three groups based on the slope of the graph at their positions.

The slopes are also listed in the figure. The sudden rise in delivered amplitude could be explained due to the resonance tracking. The authors hypothesize that at low delivered amplitudes, the BEMF of the LRA is also lower, resulting in a less accurate resonance frequency tracking. When the amplitude becomes high enough, the driver is able to estimate the resonance frequency more accurate and the delivered amplitude quickly increases. This large increase is to be expected, as a deviation of about 2.5Hz off resonance frequency can already cause a drop in LRA performance of 25 % [29]. At about 50 % set amplitude, the resonance frequency is accurately determined and the slope becomes linear again, albeit at a lower slope. In order to test this theory, the measurement was also performed at a fixed frequency of 220.8Hz. As the system is not completely matched to the resonance frequency, the LRA does not perform as well in the high amplitude ranges (about 35 % less well). The LRA does, however, show a linearity for most of the measurement, only displaying some saturation in the last two data points. At low amplitudes, the fixed amplitude is likely to be a better estimation than that of the resonance tracking, resulting in a slightly higher delivered amplitude there than the measurement with resonance tracking. Further research into the behaviour of the resonance tracking at low amplitudes, the LRA's delivered vibration amplitude at several fixed frequencies and the Q factor of the used LRAs could prove or disprove this hypothesis.

Table 6.3: All LRA characterization results.

| Set amplitude [% of maximum] | 100 | 75 | 50 | 25 |
|---|---|---|---|---|
| Amplitude [m/s$^2$] | 7.1 | 6.1 | 3.8 | 1.5 |
| Amplitude [G] | 0.72 | 0.62 | 0.39 | 0.15 |
| Measured frequency [Hz] | 223.9 | 216.2 | 218.1 | 214.0[2] |
| Rise time [ms] | 44 | 40 | 35 | 26 |
| Fall time [ms] | 44 | 40 | 33 | 26 |

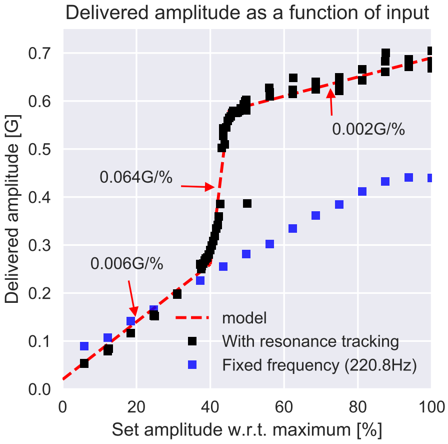


Figure 6.6: Vibration amplitude with respect to the set amplitude on the driver both with and without resonance tracking. Additionally, an estimate of the slopes in the graph of the measurements with resonance tracking is given.

An example of a sine fit can be found in Fig. 6.7. The other fits can be found in Appendix D. The y-axis is clearly out of phase with the x- and z-axes. The sine looks well fitted, aside from a few stray points. These could be due to outside influence on the accelerometer, such as someone bumping into the table on which the measurement setup was set. It should be noted that the found frequencies range from 214.0 Hz to 223.9 Hz as can be seen in Tab. 6.3. This is a significant deviation from the 235 Hz resonance frequency listed by the manufacturer, stressing the need for resonance tracking rather than simply driving the LRA at 235 Hz. It does, however, also show the large deviation of the frequency at low requested amplitudes and at higher requested amplitudes in line with the hypothesis above.

[2] This frequency could not accurately be determined. Most likely because the resonance tracking feature did not function very well for this amplitude due to a low BEMF, resulting in a drift in driving frequency during the measurement. See Fig. D.2.

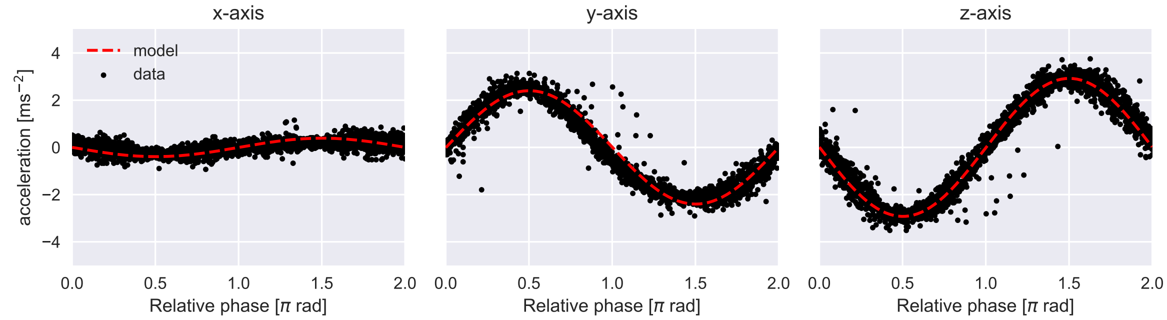


Figure 6.7: Example of a sine fit, taken at a set amplitude of 50% of the maximum. The sine has been phase folded for legibility.

Finally, an example of the rise time fit can be found in Fig. 6.8. The other fits are shown in Appendix D.

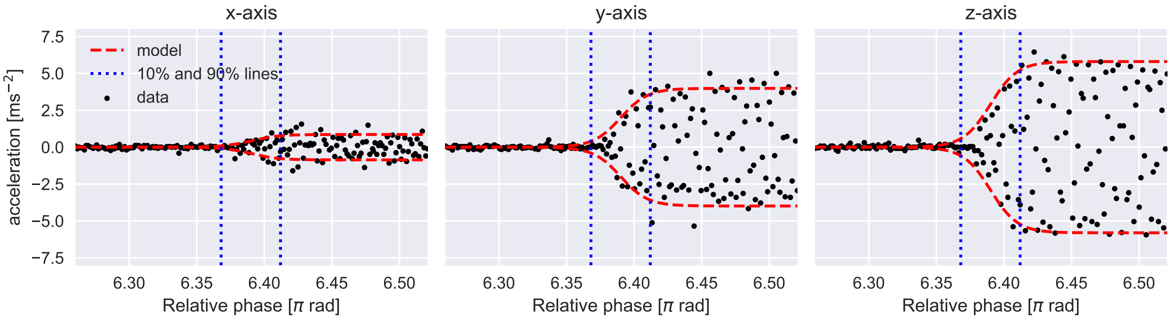


Figure 6.8: Example of the envelope fit with the 10% and 90% lines used to define the rise time. This data is taken at full power.

# 7 Prototype

For this project several different prototyping stages were used. This started with the proof of concept, which exists to validate that the intended functionality is indeed achievable, without going deeply into the design. Using the knowledge gained from this setup a first PCB was designed and ordered. In order to ensure that the final prototype works correctly two PCBs were budgeted into the project to provide room for revisions. These prototyping stages will be discussed in the sections below.

## 7.1. Proof of concept

During the proof of concept stage the group used components on breakout boards to make sure that the proposed circuit would work before money would be spent on ordering a PCB. Furthermore, initial testing was performed on latency using this circuit.

### 7.1.1. Implementation

Almost all the components that were planned to be used were available on breakout boards, which simplifies the proof of concept setup. The DRV2605L driver was available on a breakout board while the DRV2604L driver was not. Fortunately both drivers feature the exact same pinout, this means that the DRV2604L could be soldered onto a DRV2605L breakout board. In this setup a WEMOS D1 mini microcontroller was used for testing, this was chosen since the group already had one available. Firstly both LRA drivers were tested in their different operating modes while being directly attached to the microcontroller. When this testing was concluded the TCA9548A I$^2$C switch was attached between the drivers and the microcontroller. This setup was tested for functionality as well as its latency. The test setup can be found in Fig. 7.1.

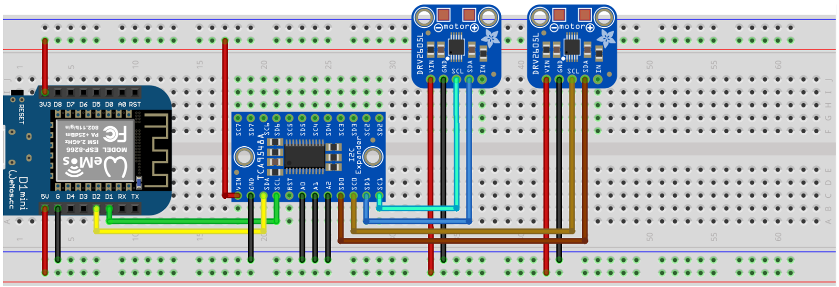


Figure 7.1: Test setup for the proof of concept.

### 7.1.2. Software design

Since this software was written for the proof of concept stage the microcontroller that would be used in the final prototype was still unknown. Therefore the software designed in this stage needed to be as universal as possible. This resulted in the choice for the Arduino platform because of its compatibility with a large number of different microcontrollers as well as being based on C/C++ which is supported by most microcontrollers.

Furthermore the Arduino platform provides easy access to the GPIO and peripherals like I$^2$C. The designed software was run on a WEMOS D1 mini, which is an ESP8266 based microcontroller.

The used Adafruit DRV2605L breakout board also came with an Arduino library [42]. This library provides access to some of the features of the driver with the notable exceptions being access to the auto calibration as well as settings for the rated and clamp voltages for the output. Using this library as a reference, functions were written to configure the desired settings and perform all the required actions.

Due to the high possibility that the microcontroller and programming language used in the final prototype would be different than the one used in this stage the code was designed to be as modular as possible. Both the LRA driver and the switch rely on I$^2$C communication, since the implementations of the I$^2$C peripheral usually differs per programming platform this communication was abstracted away into functions. Specifically the functions `write8BitReg` (which writes an 8 bit register) and `read8BitReg` (which reads an 8 bit register) for the LRA driver and `setChannel` (to select an output channel) for the I$^2$C switch. The switch specifically needs its own function for writing over I$^2$C since it only has one register to write to and thus does not expect a register address to be sent prior to the data [37].

### 7.1.3. Validation

The initial latency measurements and the force characterization were performed during the proof of concept stage. For more detail on this testing please refer to Section 6.1 and 6.2. These measurements were later used as a baseline to compare the measurements of the designed PCBs to.

## 7.2. First prototype

In this stage the first PCB was designed. This design was focused on the functionality of the different subsystems. All subsystems were routed individually and combined later, this does not make for a very efficient design in terms of size but made it relatively quick to design. The size was not a focus here since a second PCB would be designed that met these requirements.

### 7.2.1. Implementation

The PCB layout had some restrictions in terms of the placement of components. Decoupling capacitors for both the LRA drivers as well as the I$^2$C switch need to be placed as close to the IC as possible. Since the drivers have an integrated 1.8 V regulator it also had to have a filtering capacitor placed as close to the IC as possible. Since the desire is to run the I$^2$C bus at at least 400 kHz all SCL and SDA lines needed to be kept as short as possible. This is because longer traces increase the amount of parasitic capacitance.

In the case that something went wrong with the design of the I$^2$C switch jumpers were placed on the SCL and SDA lines from the microcontroller to be able to route them to driver 5 directly. Furthermore net ties were placed between the EN, TRIG/IN lines of the drivers and the RESET line of the switch and the microcontroller. This was done so that if they were routed incorrectly these lines could be disconnected and corrected by soldering a wire to this pad. Test points were considered for some of the signals however since the components used in this system are not that small pads or pins on the ICs could be probed directly with an oscilloscope probe. Therefore in view of size constraints these were omitted.

### 7.2.2. Software design

It was decided that the final prototype would use an ESP32 microcontroller and the software would be written using the ESP-IDF framework. This is the framework that is provided by the manufacturer of the ESP32. The reasoning behind these decisions can be found in Section 3.2 and 3.3. Since ESP-IDF is also based on C most of the functions written for the Arduino platform work without needing large adaptations. This is aided by the fact that the I$^2$C communication is abstracted away by functions, specifically `write8BitReg`, `read8BitReg` and `setChannel` as described above.

The functions built into the ESP-IDF often return error codes that signify what went wrong. These error codes are useful when debugging the code. However the error codes need to be read out over serial and since the SenseGlove Communication Protocol uses serial it is not desired to send these codes in the final product. Therefore the code for printing these error codes was made modular to make it easy to toggle on or off.

### 7.2.3. Validation

A test procedure was established to make sure the PCB was tested properly. It was designed such that interference of other subsystems on the PCB or effects of the routing on the PCB would be tested for and investigated. The procedure is listed below.

- The response of the I$^2$C switch is confirmed.
- The response of the drivers on the I$^2$C bus is tested.
- The change in latency with respect to the proof of concept is investigated.
  - Due to using ESP-IDF instead of Arduino.
  - Due to a faster I$^2$C bus.

While assembling the PCB it was discovered that the pull-up resistors on the I$^2$C bus connecting the I$^2$C switch, as well as other components like the amplifier for the Lofelt circuit, to the microcontroller were missing. Initially these resistors were placed in the design for the LRA circuit, however with the addition of more components on the bus they were removed from the LRA schematic in order to be moved to the schematic for the microcontroller. This however was forgotten. Therefore resistors were placed on the jumpers from in the LRA circuit to properly pull up the lines without needing to order a new PCB.

When the PCB was tested it became apparent that the fifth driver responded to all commands regardless of what channel the I$^2$C switch was set to. This was found to be because the pull-up resistors that were soldered onto the jumpers accidentally connected the fifth driver directly to the I$^2$C bus coming from the microcontroller. Luckily this was solved by moving the pull-up resistors to the other side of the jumper.

## 7.3. Final prototype

Since the first prototype of the LRA PCB proved to function as intended aside from the missing pull-up resistors, most of the schematic could be re-used and the design of the final prototype was focused mostly on minimizing the required area with the constraint of board dimensions.

### 7.3.1. Implementation

Due to the large size of the PCB of the first prototype it was decided to create a 4 layer PCB for the final prototype. Furthermore this PCB would be split up into two parts. One part that would be mounted on top of the wrist with the other on the bottom of the wrist. Both sides also carry one cell of the battery, this is done such that the weight is distributed evenly. This does mean that only one side of the PCB can contain components, otherwise they would interfere with the battery. The dimensions for these PCBs was taken slightly larger than the size of the used battery cells and came out at 70 by 40 mm. Furthermore the shape of the space available for each component was different, the expected layout can be seen in Fig. 3.3. Since the layout for the LRA circuit was more or less square for the first prototype and the fact that the I$^2$C switch needed to be moved back the layout had to be created from scratch. However, due to the experience gained when working on the first prototype, the design for the final version could be made more quickly.
As can be seen in Fig. 3.3 the I$^2$C buses coming from the switch have to be routed through the circuitry required for the finger force feedback. Therefore, the decision was made to route the data lines for the LRA circuit on the top layer while the finger force feedback lines would be routed on the bottom layer as much as possible, passing underneath the power and ground planes of the LRA subsystem.

### 7.3.2. Software design

At this point most of the software has been written and tested. however the most important thing that still needs to be implemented is the integration of the SenseGlove Communication Protocol.

### 7.3.3. Validation

Unfortunately at time of writing the PCB for the final prototype has not been assembled yet. The PCB was ordered on June 18 from Eurocircuits. Due to the thesis deadline of June 21 there was not enough time to solder all the components and test the PCB. This will however be done during the time between the thesis deadline and the defense such that the final prototype can be demonstrated at the defense.

# 8 Discussion

One of the main limiting factors in this project was the latency. Keeping this to a minimum is essential in a virtual reality environment to maintain immersion. This project has a requirement to have latency of 40 ms in the electrical part of the system. Most of this latency budget will be taken up by the communication over Bluetooth. The time taken to select the correct driver and send an amplitude is about 0.8 ms, while the time taken for the microcontroller to send an amplitude to each driver is 1.433 ms as shown in Section 6.1. A major part of this delay is thus waiting for the drivers to generate an output, however, this can occur in parallel to the other drivers. Since the time to send an amplitude includes the time to send an instruction the combined latency for all five drivers can then be calculated as $t_{\text{total}} = \frac{4}{5} \cdot t_{\text{exec}} + t_{\text{switch}} = \frac{4}{5} \cdot 1.433 + 0.8 = 1.9$ ms. This value is below the expected latency of 2.5 ms and is a minor part of the 40 ms latency budget.
Aside from the electrical latency, the latency of the LRAs was also characterized in terms of their rise time, as stated in Section 6.2. Therefore, an estimation of the total latency including mechanical latency can be made. This definition of latency differs from the one described in the requirements as that does not include the mechanical latency. It is, however, compelling to look into the mechanical latency, as it helps the designer understand the actual haptic experience of the user. The 10 % to 90 % rise time of the LRA at maximum vibration strength was estimated to be 44 ms. When adding this to the estimated latency of the Bluetooth link (7 ms), see Section 3.4, and the achieved latency of the finger vibrotactile feedback subsystem calculated above, the total delay between the sending of a signal at the computer and the activation of the LRA is about 53 ms for maximum vibration strength. This is reduced up to almost 20 ms for smaller amplitude rises and falls, as the 10 % to 90 % rise time for a quarter of the strength is already reduced to 26 ms. This means that the time between the computer software deciding haptic feedback needs to be sent and the perception of mechanical movement of the LRA subsystem will mostly be between 35 ms and 50 ms. The 53 ms limit will most likely not often be reached, as this only occurs when the LRA switches from being completely motionless to maximum strength and vice versa, whereas smaller differences in amplitude have a lower rise and fall time. Additionally, the user will already be able to detect a change in vibration strength before the entire 10 % to 90 % rise or fall has been completed. The reaction of the vibrotactile feedback system therefore occurs in a range where people are not able to perceive the latency in the feedback [43], provided there is no large latency on the software side. The 50 ms limit provides for a better immersion [44].

The other main goal of this project is that the SoftGlove provides better vibrotactile feedback than the system that is used on the current SenseGlove. Better vibrotactile feedback is hard to define since it is inherently subjective. However the designed system has a larger specified vibration strength than the ERM of the current SenseGlove, 1.5 G instead of 0.8 G. It should be noted that the authors only measured an amplitude of 0.72 G or a peak to peak acceleration of 1.44 G. Additionally this vibration occurs perpendicular to the finger instead of the tangential vibration of the ERM. Since the finger vibrotactile feedback is among other things meant for button presses having a vibration perpendicular to the finger is more realistic. Finally the used LRAs have a rise and fall time of at most 44 ms 6.3 while utilizing the automatic braking and overdrive feature of the DRV2604L. This is definitely an improvement over the 142 and 61 ms respectively for the current ERM [6].

# 9 Conclusions, recommendations and future work

## 9.1. Conclusions

A system has been designed that features five linear resonant actuators to provide finger vibrotactile feedback. This system is integrated together with the finger force feedback as well as the vibrotactile feedback in the hand palm. The characteristics of the designed system consists of a latency of 1.9 ms, a specified vibration force of 1.5 G and a 10 % to 90 % rise and fall time of 44 ms from rest to maximum amplitude 6.3. Furthermore, the LRAs are placed on the intermediate phalanges, since the thumb does not have an intermediate phalanx it is placed on the proximal phalanx [19] on the back of the hand. This provides a good compromise of interference with the ability to grab an object, while still providing satisfying feedback. Finally the chosen LRA has a specified resonance frequency of 235 Hz meaning that it is within the range in which the human skin is most sensitive to vibrations [19].

It can thus be said that all mandatory requirements satisfied, provided that the designed battery protection circuit on the final prototype functions as expected. As for the cost factors, the latency of the LRA subsystem is relatively low with a latency of 1.9 ms. Furthermore the LRAs have with 1.5 W peak a relatively low power consumption. With the PCB size of 70 by 40 mm the team considers cost factor 2 satisfied. LRAs have a longer lifespan than the ERMs used in the current SenseGlove [45] thereby satisfying cost factor 5. Lastly, the size of the final product is small and the vibration frequency of the LRAs is within the most sensitive range of the hand for both men and women [19], thus also satisfying cost factor 6. Finally, the drivers of the LRA can be controlled through requesting an amplitude level over I$^2$C, which is compatible with the way the amplitudes are being sent over the SenseGlove communication protocol, in line with stretch goal 1.

## 9.2. Recommendations and future work

During this project the focus was not placed upon to the best placement of the LRAs on the hand. This will most likely play a major role in the effectiveness of the presented vibrations, leading to an overall better product. Apart from the placement some more research could be done on the comfort and performance of the different available LRAs. In this report only the G0832022D by Jinlong Machinery was characterized. However the same test setups could be used to investigate other LRAs by both Jinlong Machinery as well as Precision Microdrives or other manufacturers if they start producing LRAs.

The auto calibration feature of the LRA drivers was not successfully implemented during the project. With more time available for troubleshooting or by using the newer version of the LRA driver (the DRV2624 or DRV2625) the cause of this failure can be found and rectified. Furthermore the effect of this feature can be researched by for example making use of the force characterization test setup as explained in Section 6.2. Additionally the ERM used in the current SenseGlove can be tested on the same setup to provide a head-to-head comparison of the employed vibration motors.

Unfortunately no time remained in the project to implement functions to facilitate the loading of waveforms into the integrated RAM of the DRV2604L driver. Implementing this feature would reduce the amount of data to be send over the wireless link during operation since only the desired waveform needs to be requested instead of sending the amplitudes of the waveform individually each time. This would however also require some adaptations to the SenseGlove Communication Protocol since it currently has no way to preload a certain waveform. The Lofelt subsystem, as discussed in [2], could likely also benefit from the same feature if this were added to the protocol.

# A

# General appendix

## A.1. Schematic

### A.1.1. Module overview

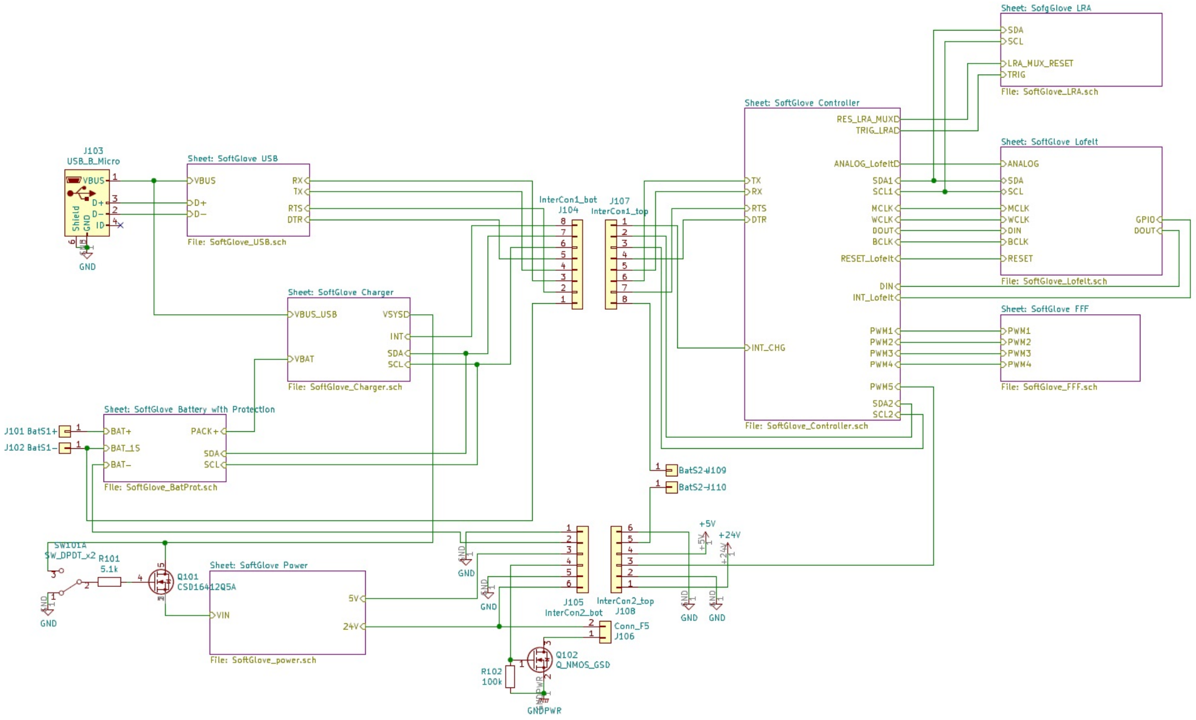


Figure A.1: Schematics of the complete system.

## A.1.2. Battery charger

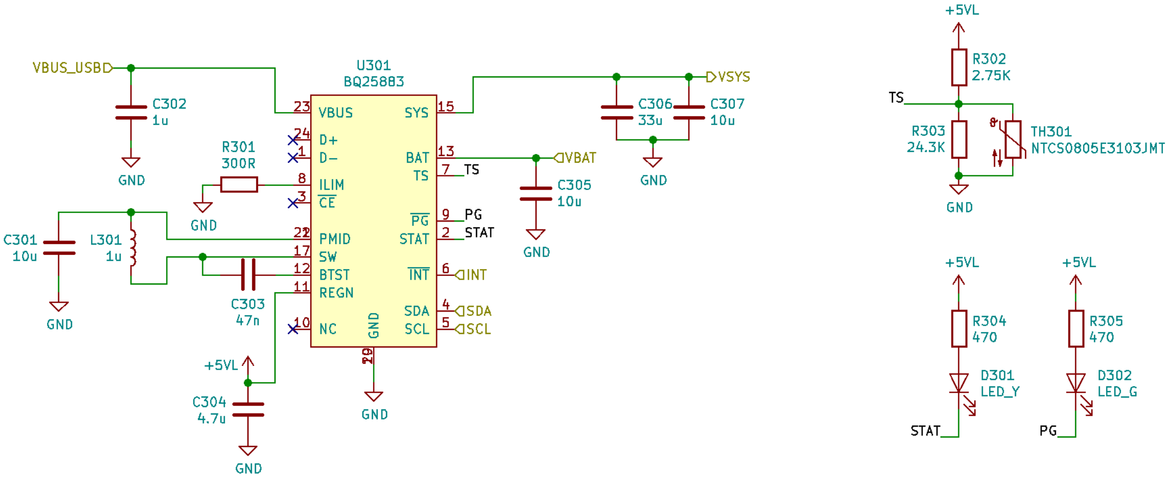


Figure A.2: Schematics of the battery charger.

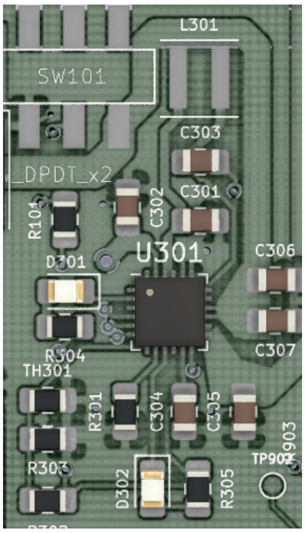


Figure A.3: PCB design of the battery charger.

### A.1.3. Battery protection and USB

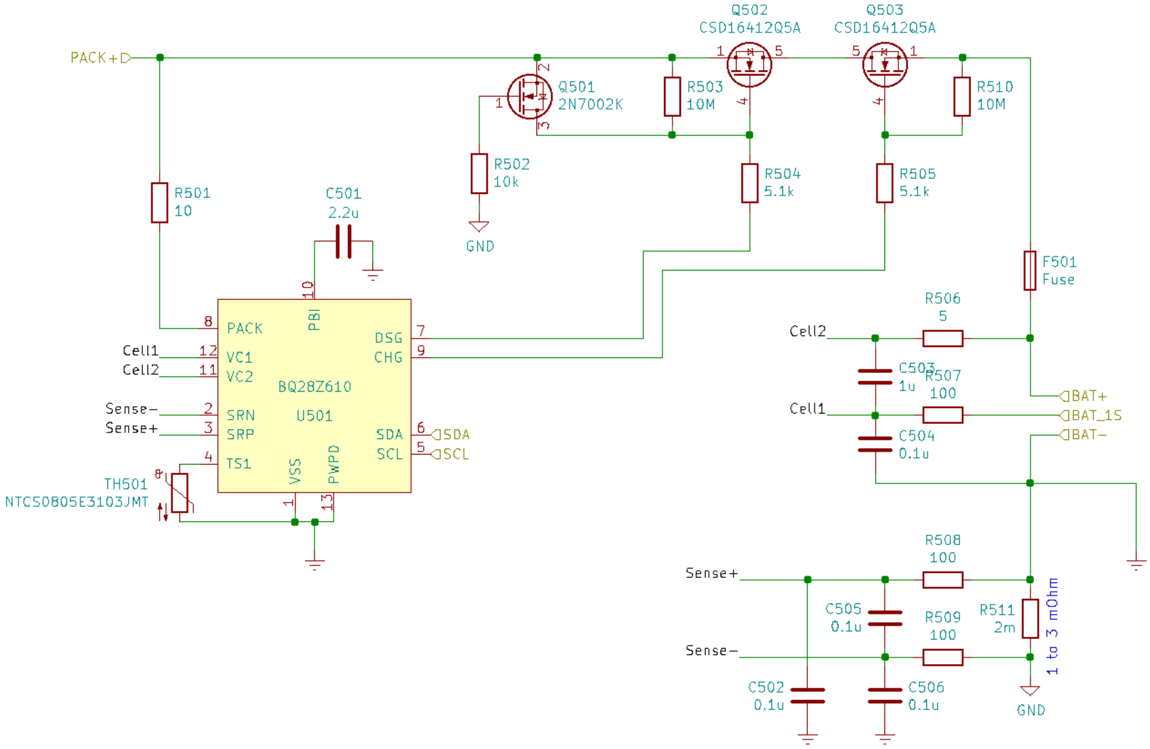


Figure A.4: Schematics of the battery protection.

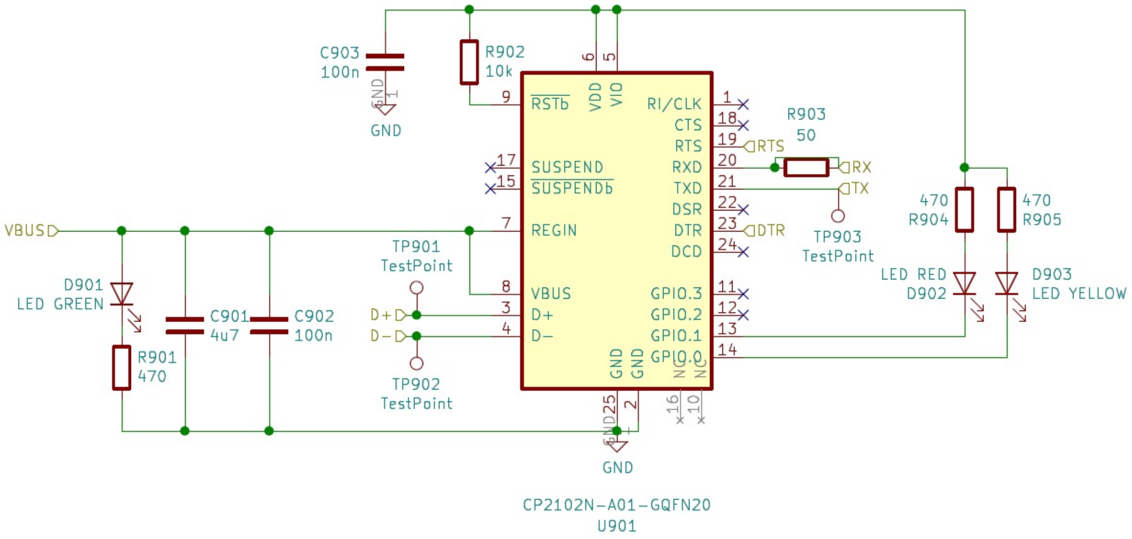


Figure A.5: Schematics of the USB to serial.

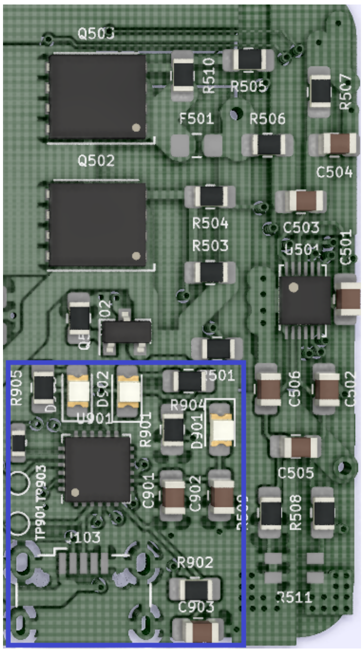


Figure A.6: PCB design of the battery protection and USB to serial design in the blue box.

### A.1.4. ESP Layout

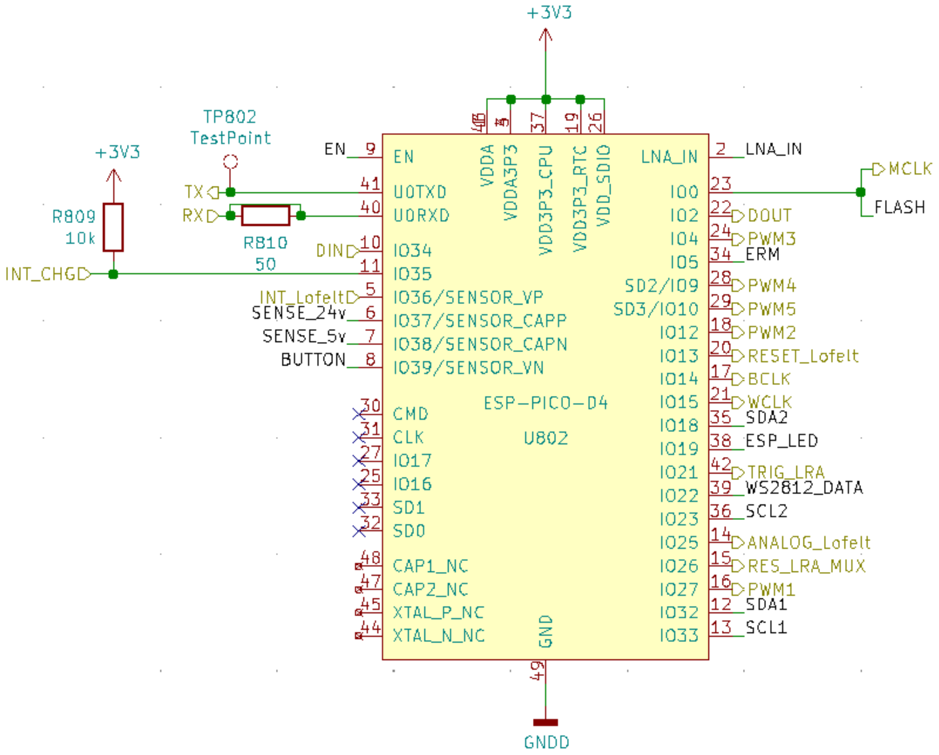


Figure A.7: The layout of the ESP with all pin connections.

### A.1.5. ESP Schematics

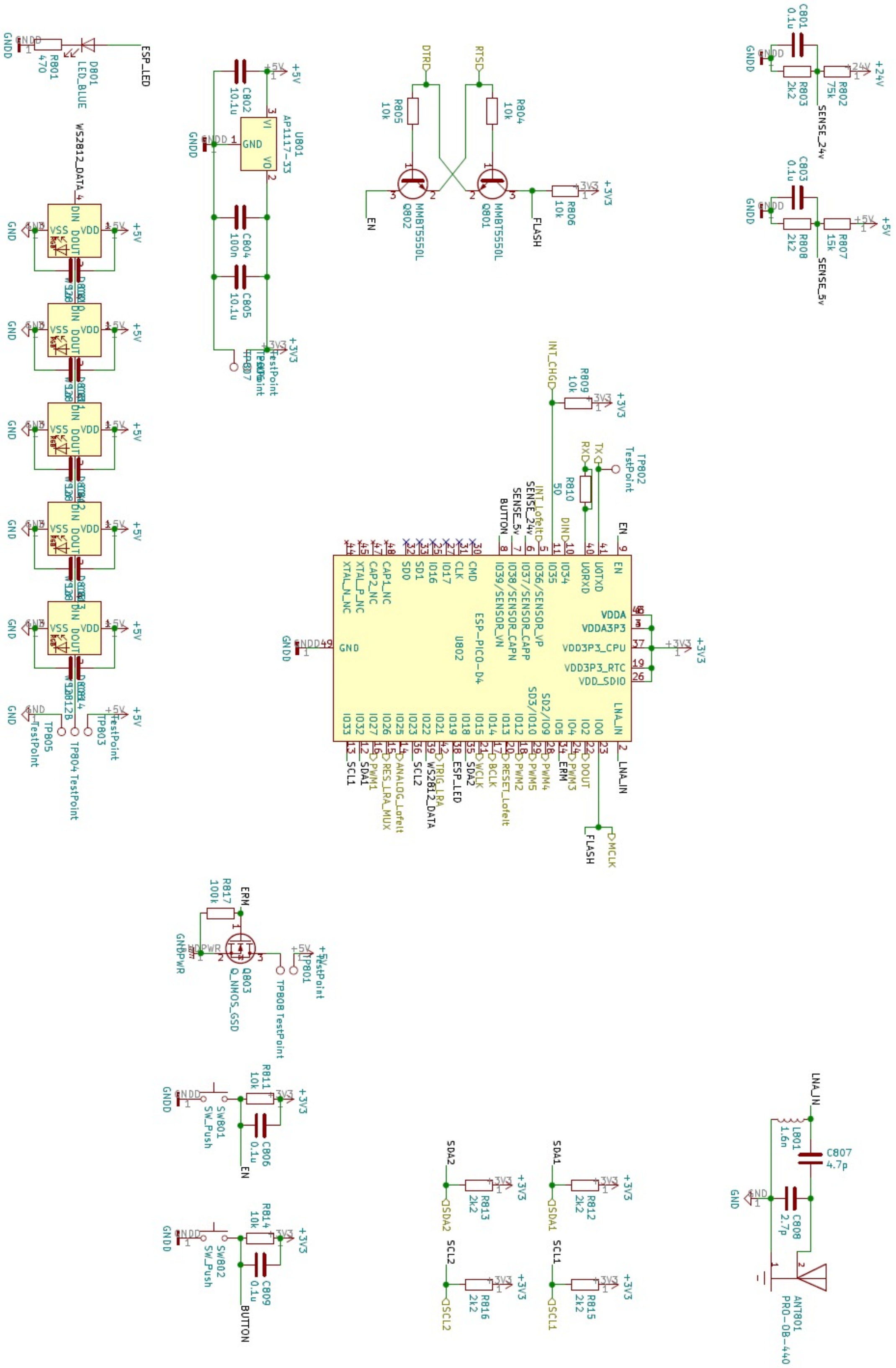
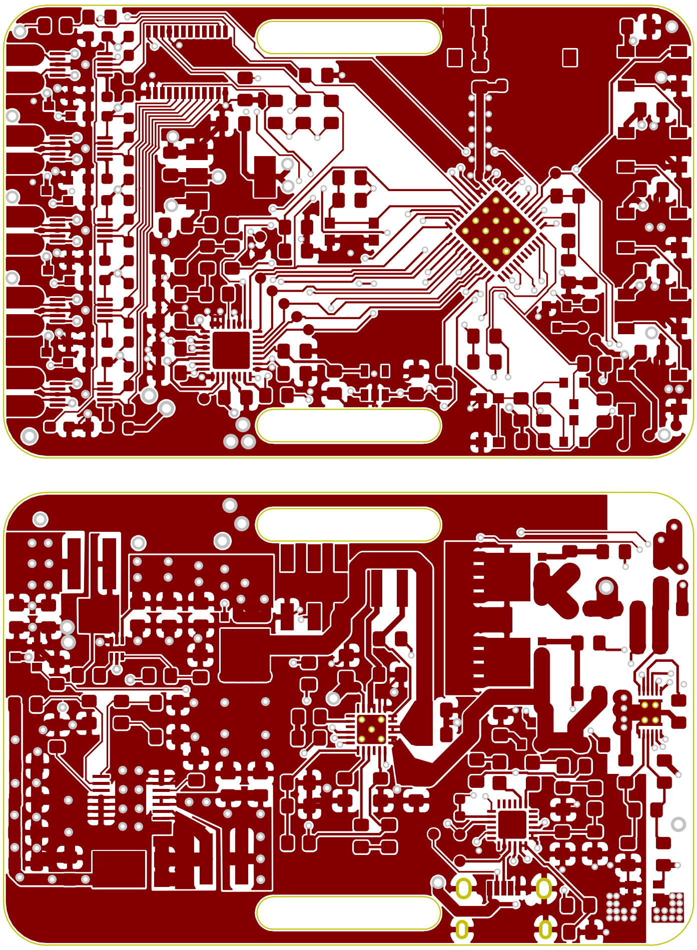


Figure A.8: Schematics of ESP (confidential).

## A.2. PCB Structure of all layers

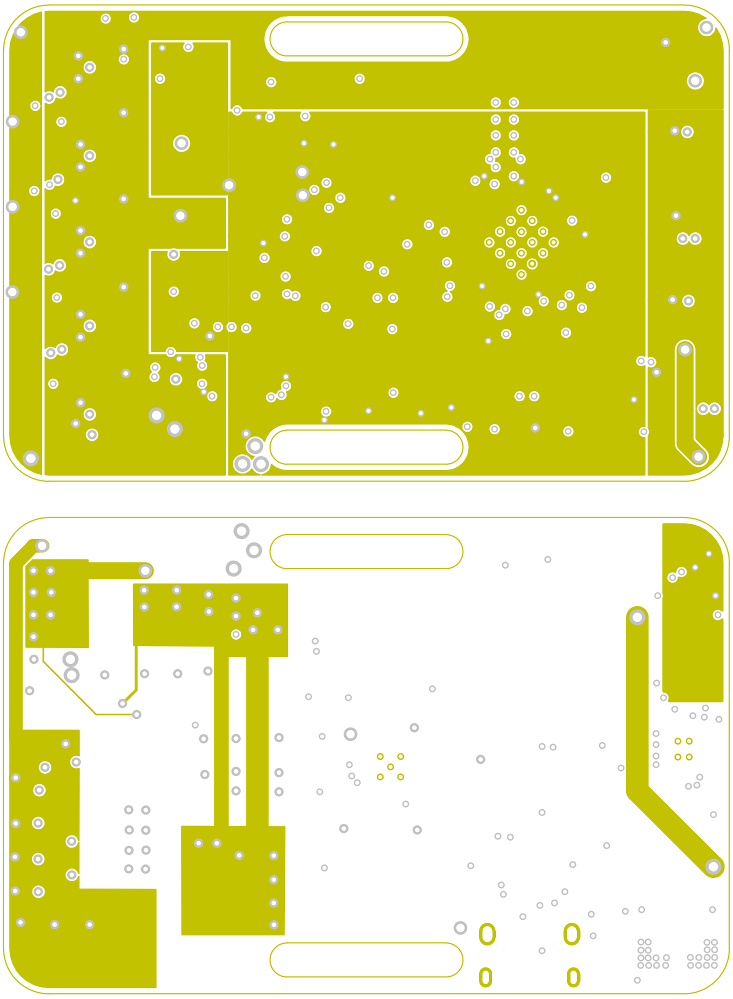
### A.2.1. Copper layer 1

(confidential)



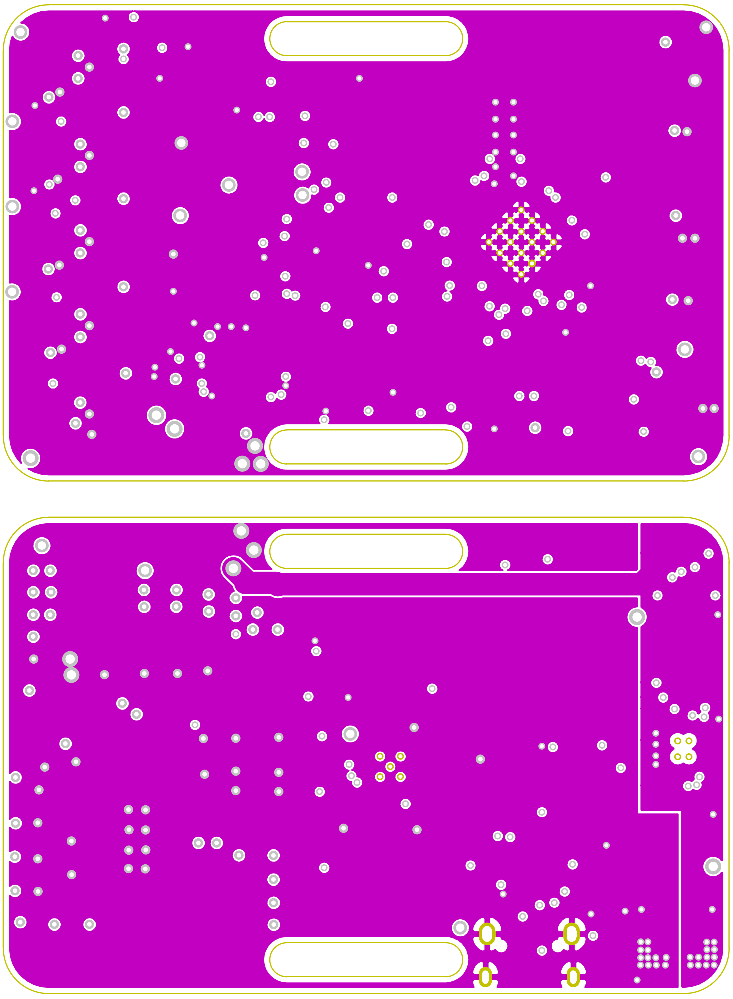
### A.2.2. Copper layer 2

(confidential)



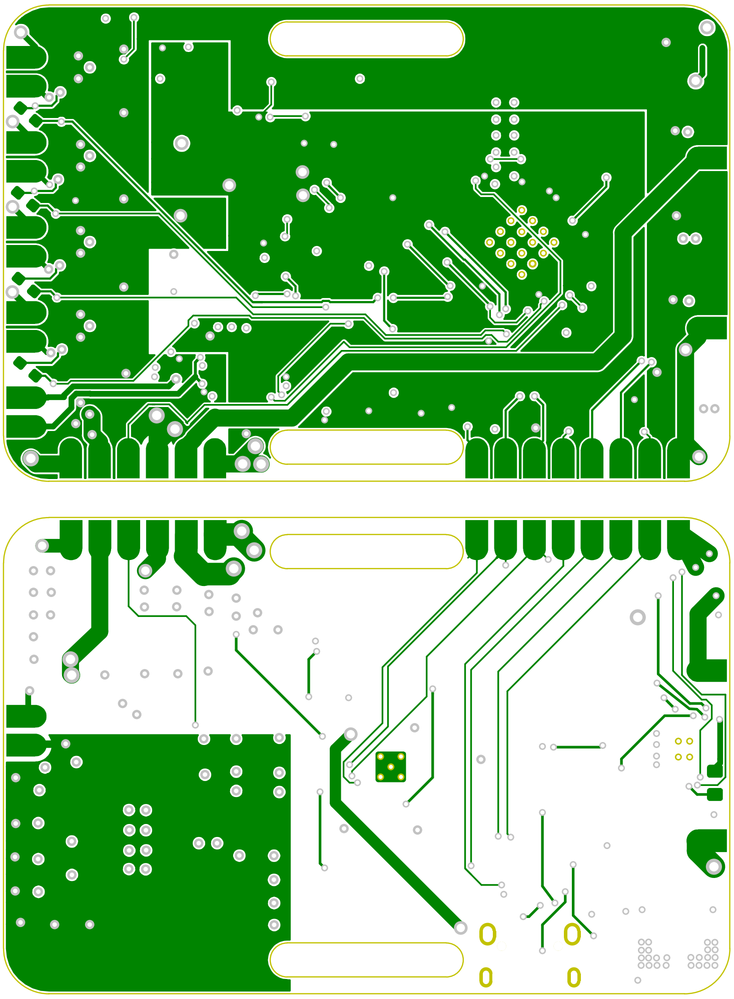
### A.2.3. Copper layer 3

(confidential)



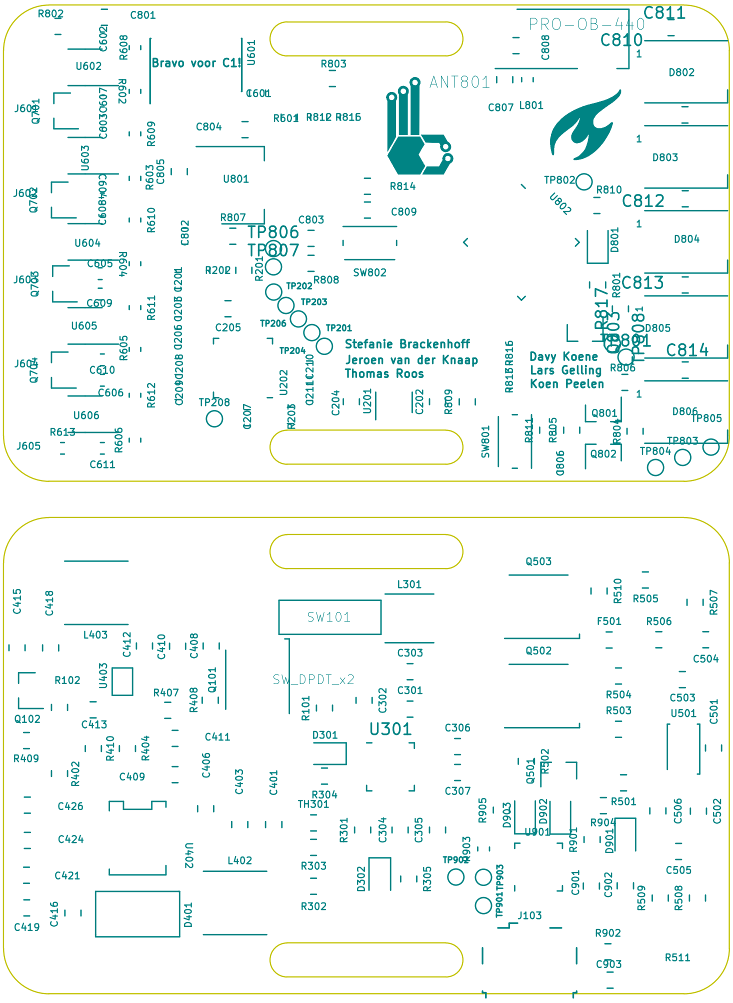
### A.2.4. Copper layer 4

(confidential)



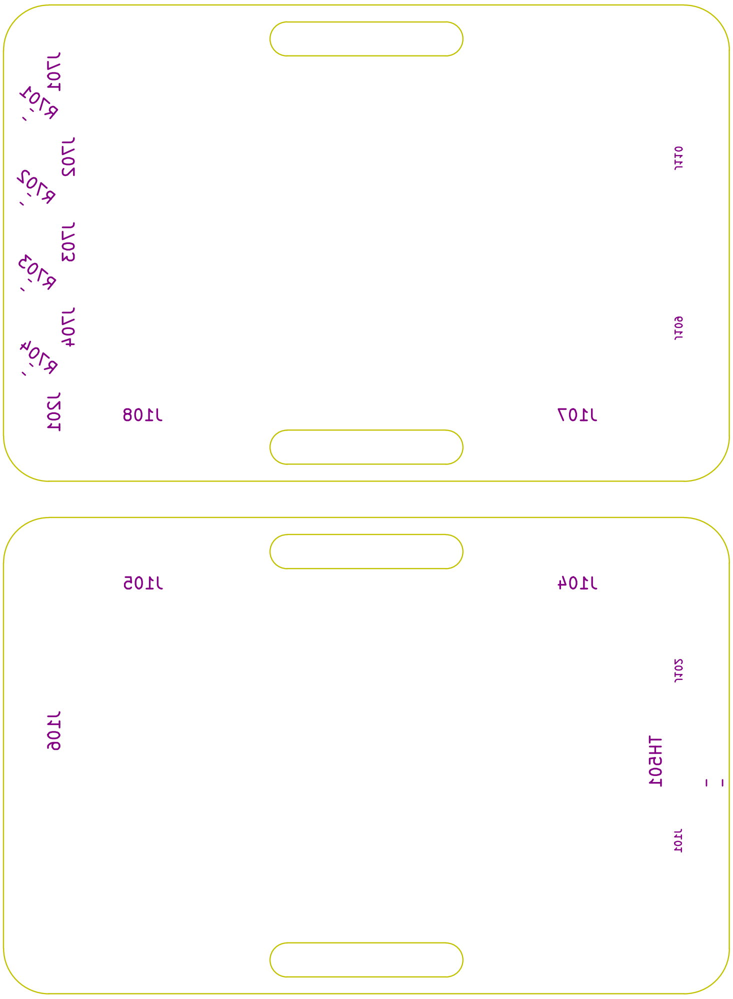
### A.2.5. Silkscreen top

(confidential)



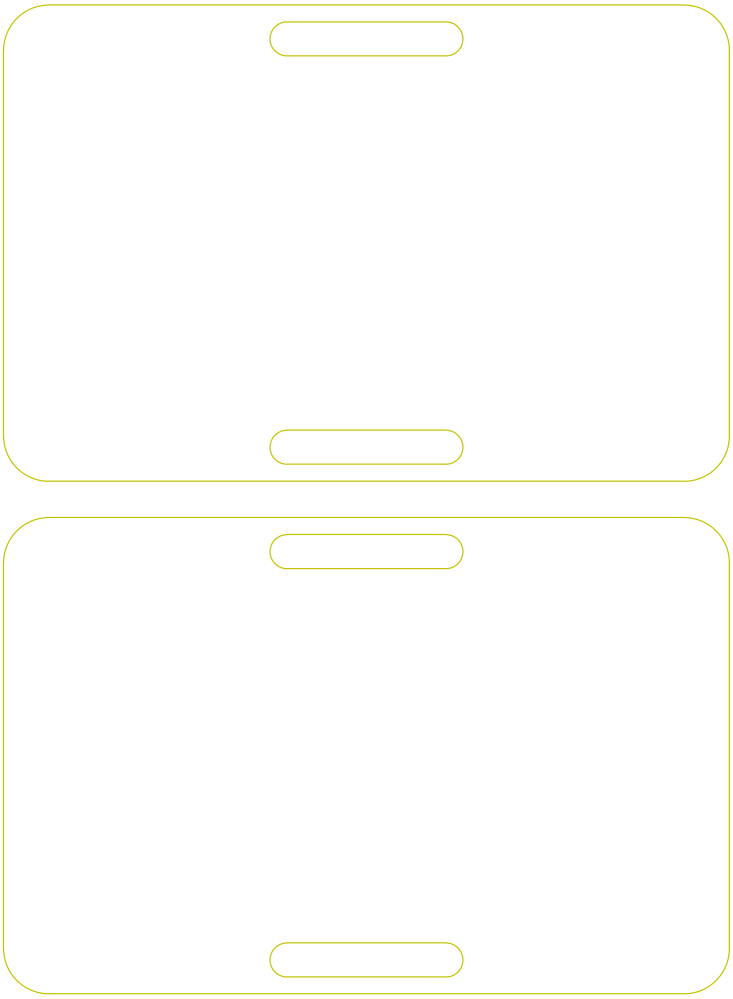
### A.2.6. Silkscreen bottom

(confidential)



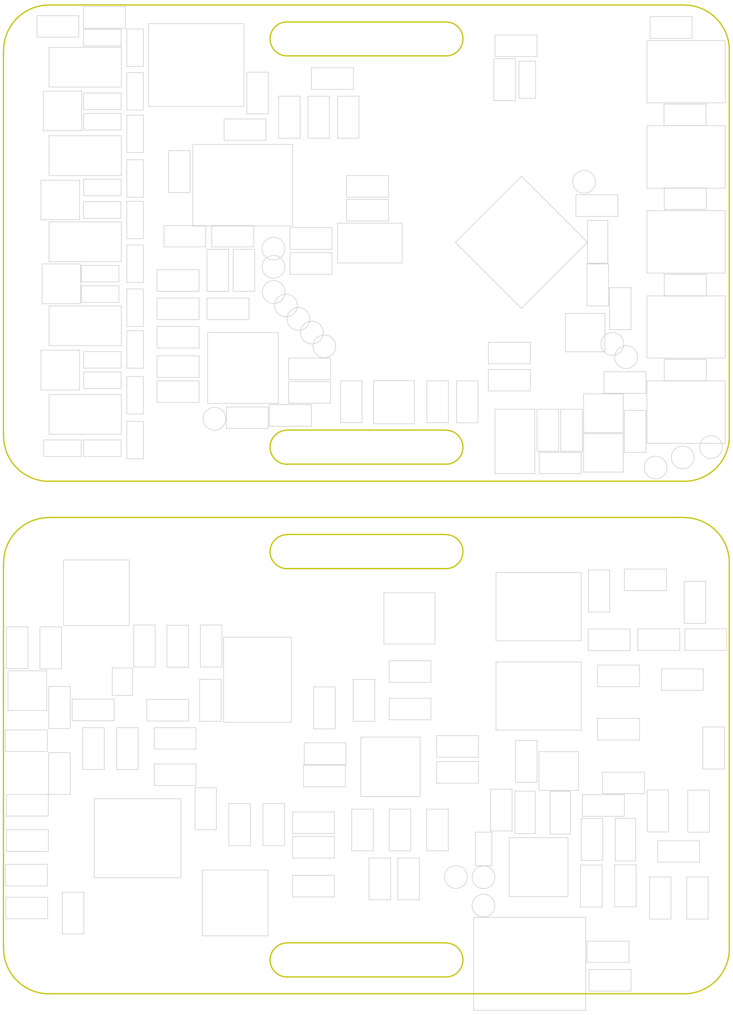
### A.2.7. Edges and routing

(confidential)



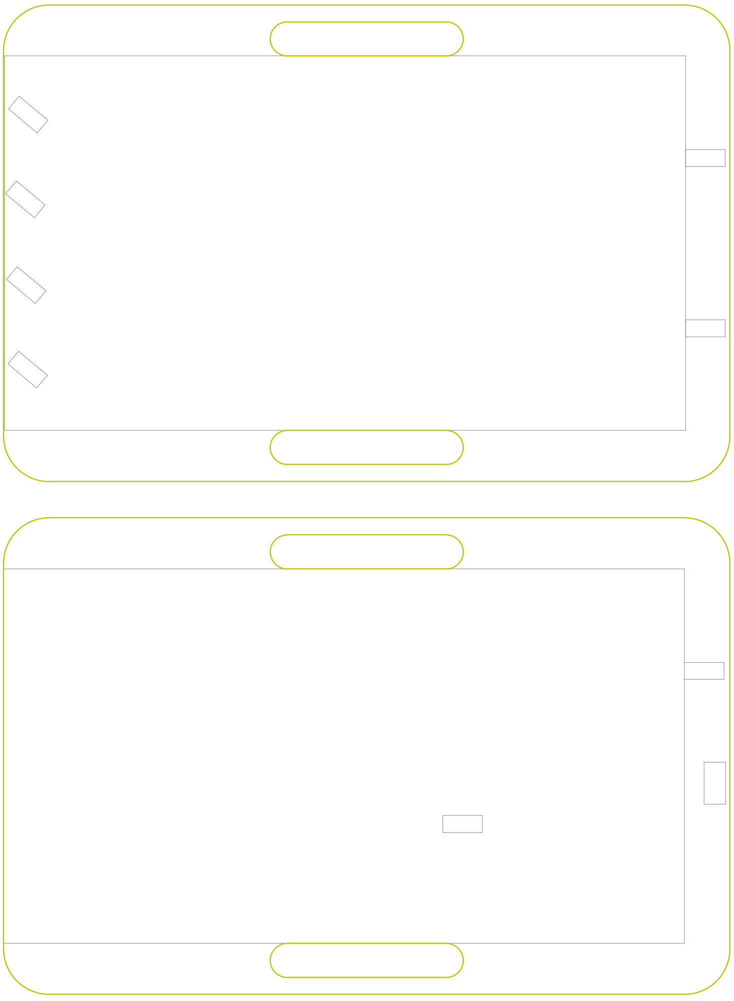
### A.2.8. Component placement top

(confidential)



### A.2.9. Component placement bottom

(confidential)



# A.3. Assignments

## A.3.1. Old assignment

### Sense Glove: Soft Glove Prototyping

Bachelor Final Project

**Company:**
At Sense Glove we develop a VR glove that translates the hands of a user to the virtual world: the Senseglove. The capabilities of the Senseglove allow a user to handle virtual objects the same as real objects. Capabilities such as per finger force- and vibrotactile feedback in addition to accurate self-contained hand tracking. The Senseglove is used in training simulators for car mechanics in a digital factory, VR CAD, proxy robotics and many more. Currently, Sense Glove has produced and sold their initial development kit. In addition to selling the Senseglove, Sense Glove helps companies to integrate interactable physics into existing VR environments. With the current development kits targeting the business-to-business market; a consumer version will be designed.

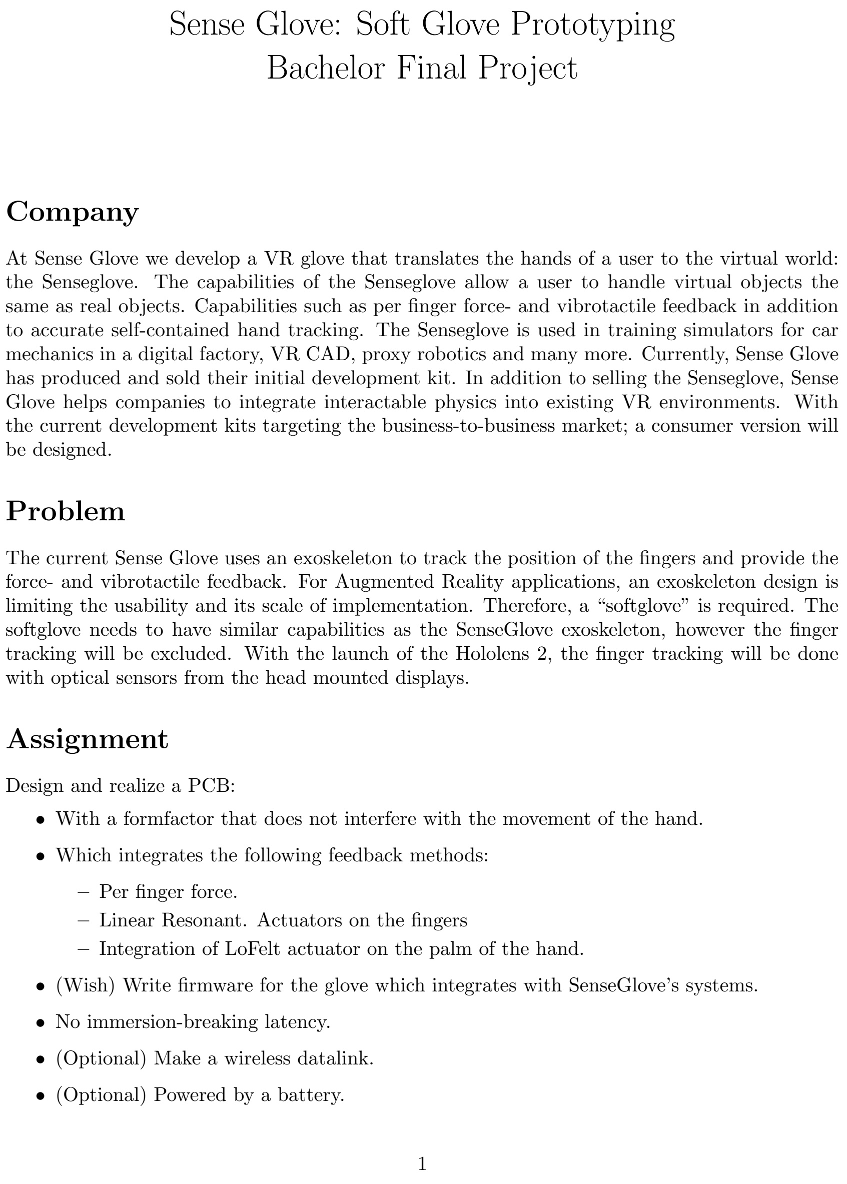
**Problem:**
The current Sense Glove uses an exoskeleton to track the position of the fingers and provide the force- and vibrotactile feedback. For Augmented Reality applications, an exoskeleton design is limiting the usability and its scale of implementation. Therefore, a "softglove" is required. The softglove needs to have similar capabilities as the SenseGlove exoskeleton, however the finger tracking will be excluded. With the launch of the Hololens 2, the finger tracking will be done with optical sensors from the head mounted displays.

**Assignment**

1. Design and realize a semi-flex PCB for the softglove, which integrates
   a. Per finger force feedback
   b. Linear Resonant Actuators in the fingertips
   c. Integration of LoFelt haptic drivers on the palm of the hand
2. Write firmware for the PCB, which can communicate to a PC through USB.
3. (Optional) Make it wireless through Bluetooth.

Sense Glove | MAKING VIRTUAL REALITY REAL

### A.3.2. New assignment



## Sense Glove: Soft Glove Prototyping
## Bachelor Final Project

### Company

At Sense Glove we develop a VR glove that translates the hands of a user to the virtual world: the Senseglove. The capabilities of the Senseglove allow a user to handle virtual objects the same as real objects. Capabilities such as per finger force- and vibrotactile feedback in addition to accurate self-contained hand tracking. The Senseglove is used in training simulators for car mechanics in a digital factory, VR CAD, proxy robotics and many more. Currently, Sense Glove has produced and sold their initial development kit. In addition to selling the Senseglove, Sense Glove helps companies to integrate interactable physics into existing VR environments. With the current development kits targeting the business-to-business market; a consumer version will be designed.

### Problem

The current Sense Glove uses an exoskeleton to track the position of the fingers and provide the force- and vibrotactile feedback. For Augmented Reality applications, an exoskeleton design is limiting the usability and its scale of implementation. Therefore, a "softglove" is required. The softglove needs to have similar capabilities as the SenseGlove exoskeleton, however the finger tracking will be excluded. With the launch of the Hololens 2, the finger tracking will be done with optical sensors from the head mounted displays.

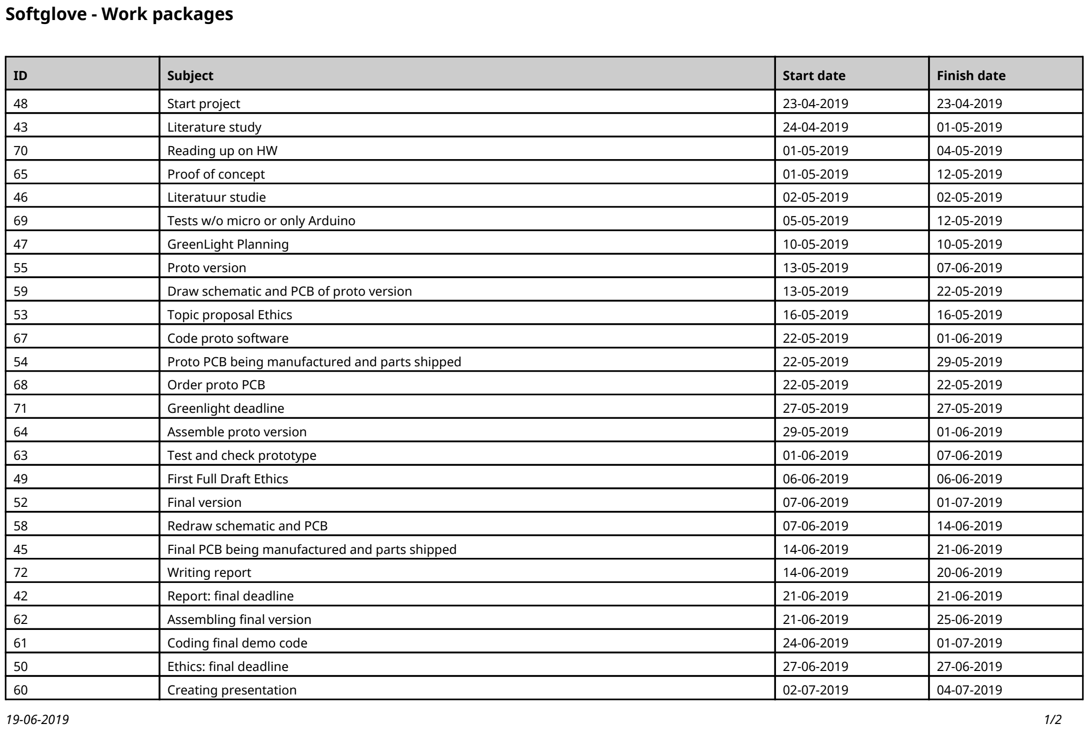
### Assignment

Design and realize a PCB:

- With a formfactor that does not interfere with the movement of the hand.
- Which integrates the following feedback methods:
  - Per finger force.
  - Linear Resonant. Actuators on the fingers
  - Integration of LoFelt actuator on the palm of the hand.
- (Wish) Write firmware for the glove which integrates with SenseGlove's systems.
- No immersion-breaking latency.
- (Optional) Make a wireless datalink.
- (Optional) Powered by a battery.

# A.4. Planning

## A.4.1. New assignment



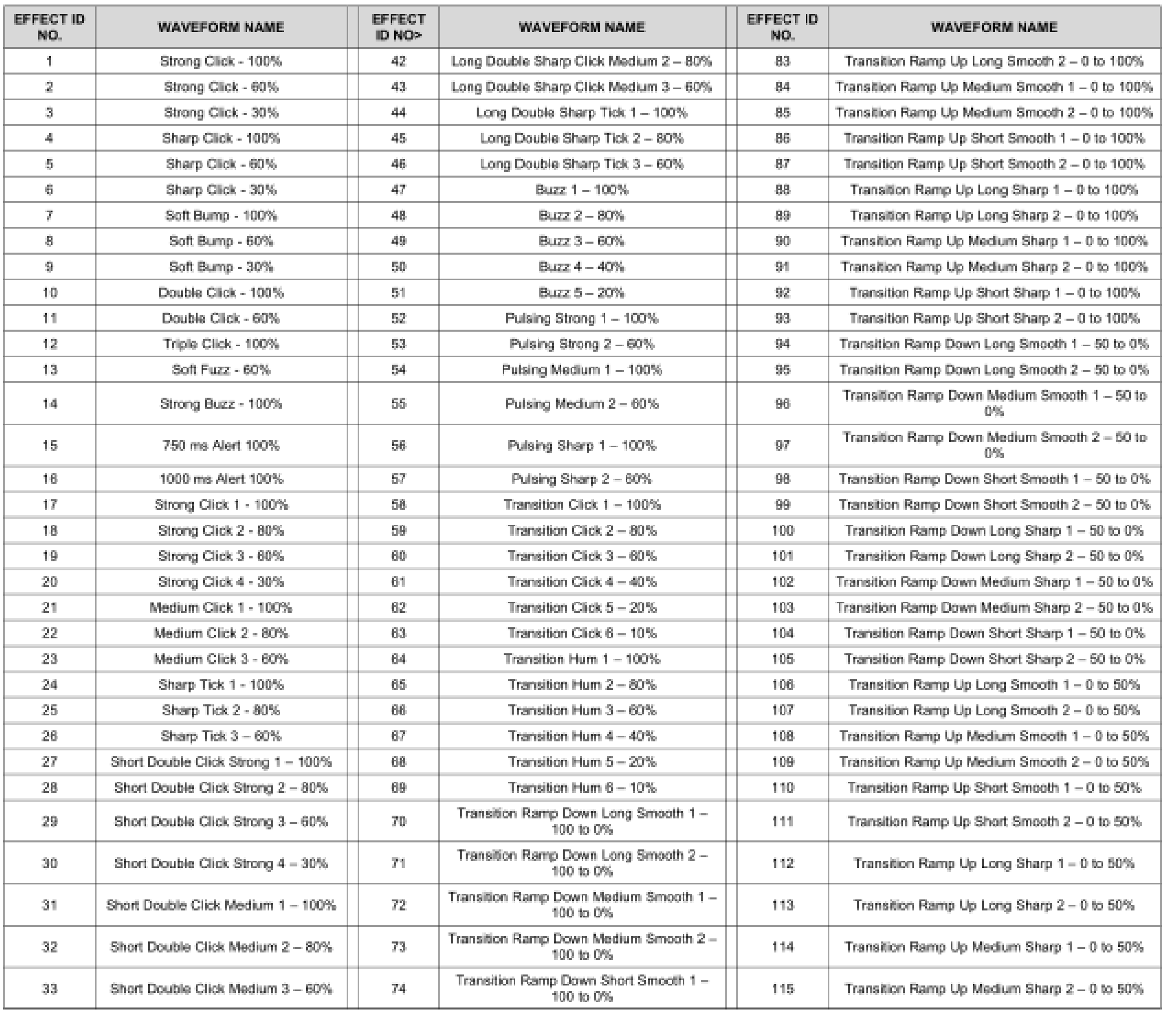
**Softglove - Work packages**

| ID | Subject | Start date | Finish date |
|---|---|---|---|
| 48 | Start project | 23-04-2019 | 23-04-2019 |
| 43 | Literature study | 24-04-2019 | 01-05-2019 |
| 70 | Reading up on HW | 01-05-2019 | 04-05-2019 |
| 65 | Proof of concept | 01-05-2019 | 12-05-2019 |
| 46 | Literatuur studie | 02-05-2019 | 02-05-2019 |
| 69 | Tests w/o micro or only Arduino | 05-05-2019 | 12-05-2019 |
| 47 | GreenLight Planning | 10-05-2019 | 10-05-2019 |
| 55 | Proto version | 13-05-2019 | 07-06-2019 |
| 59 | Draw schematic and PCB of proto version | 13-05-2019 | 22-05-2019 |
| 53 | Topic proposal Ethics | 16-05-2019 | 16-05-2019 |
| 67 | Code proto software | 22-05-2019 | 01-06-2019 |
| 54 | Proto PCB being manufactured and parts shipped | 22-05-2019 | 29-05-2019 |
| 68 | Order proto PCB | 22-05-2019 | 22-05-2019 |
| 71 | Greenlight deadline | 27-05-2019 | 27-05-2019 |
| 64 | Assemble proto version | 29-05-2019 | 01-06-2019 |
| 63 | Test and check prototype | 01-06-2019 | 07-06-2019 |
| 49 | First Full Draft Ethics | 06-06-2019 | 06-06-2019 |
| 52 | Final version | 07-06-2019 | 01-07-2019 |
| 58 | Redraw schematic and PCB | 07-06-2019 | 14-06-2019 |
| 45 | Final PCB being manufactured and parts shipped | 14-06-2019 | 21-06-2019 |
| 72 | Writing report | 14-06-2019 | 20-06-2019 |
| 42 | Report: final deadline | 21-06-2019 | 21-06-2019 |
| 62 | Assembling final version | 21-06-2019 | 25-06-2019 |
| 61 | Coding final demo code | 24-06-2019 | 01-07-2019 |
| 50 | Ethics: final deadline | 27-06-2019 | 27-06-2019 |
| 60 | Creating presentation | 02-07-2019 | 04-07-2019 |

*19-06-2019* *1/2*

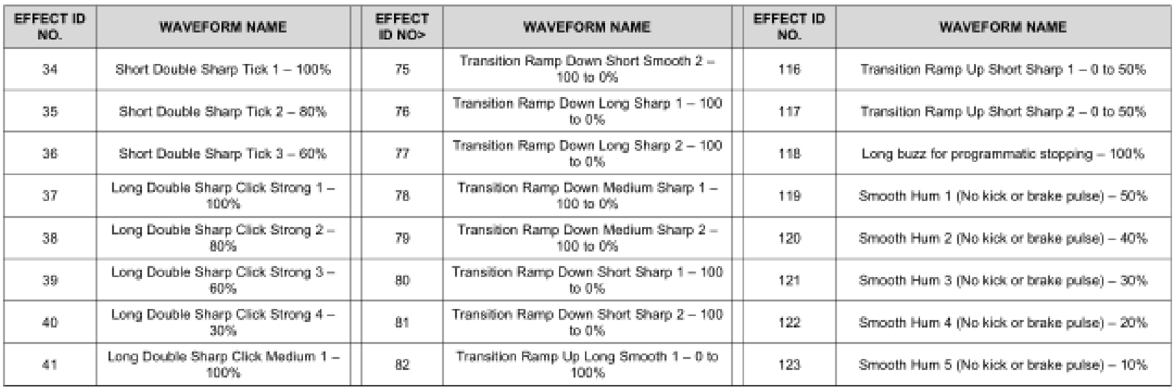
# B

# TouchSense 2200 Library



| EFFECT ID NO. | WAVEFORM NAME | EFFECT ID NO> | WAVEFORM NAME | EFFECT ID NO. | WAVEFORM NAME |
|---|---|---|---|---|---|
| 1 | Strong Click - 100% | 42 | Long Double Sharp Click Medium 2 – 80% | 83 | Transition Ramp Up Long Smooth 2 – 0 to 100% |
| 2 | Strong Click - 60% | 43 | Long Double Sharp Click Medium 3 – 60% | 84 | Transition Ramp Up Medium Smooth 1 – 0 to 100% |
| 3 | Strong Click - 30% | 44 | Long Double Sharp Tick 1 – 100% | 85 | Transition Ramp Up Medium Smooth 2 – 0 to 100% |
| 4 | Sharp Click - 100% | 45 | Long Double Sharp Tick 2 – 80% | 86 | Transition Ramp Up Short Smooth 1 – 0 to 100% |
| 5 | Sharp Click - 60% | 46 | Long Double Sharp Tick 3 – 60% | 87 | Transition Ramp Up Short Smooth 2 – 0 to 100% |
| 6 | Sharp Click - 30% | 47 | Buzz 1 – 100% | 88 | Transition Ramp Up Long Sharp 1 – 0 to 100% |
| 7 | Soft Bump - 100% | 48 | Buzz 2 – 80% | 89 | Transition Ramp Up Long Sharp 2 – 0 to 100% |
| 8 | Soft Bump - 60% | 49 | Buzz 3 – 60% | 90 | Transition Ramp Up Medium Sharp 1 – 0 to 100% |
| 9 | Soft Bump - 30% | 50 | Buzz 4 – 40% | 91 | Transition Ramp Up Medium Sharp 2 – 0 to 100% |
| 10 | Double Click - 100% | 51 | Buzz 5 – 20% | 92 | Transition Ramp Up Short Sharp 1 – 0 to 100% |
| 11 | Double Click - 60% | 52 | Pulsing Strong 1 – 100% | 93 | Transition Ramp Up Short Sharp 2 – 0 to 100% |
| 12 | Triple Click - 100% | 53 | Pulsing Strong 2 – 60% | 94 | Transition Ramp Down Long Smooth 1 – 50 to 0% |
| 13 | Soft Fuzz - 60% | 54 | Pulsing Medium 1 – 100% | 95 | Transition Ramp Down Long Smooth 2 – 50 to 0% |
| 14 | Strong Buzz - 100% | 55 | Pulsing Medium 2 – 60% | 96 | Transition Ramp Down Medium Smooth 1 – 50 to 0% |
| 15 | 750 ms Alert 100% | 56 | Pulsing Sharp 1 – 100% | 97 | Transition Ramp Down Medium Smooth 2 – 50 to 0% |
| 16 | 1000 ms Alert 100% | 57 | Pulsing Sharp 2 – 60% | 98 | Transition Ramp Down Short Smooth 1 – 50 to 0% |
| 17 | Strong Click 1 - 100% | 58 | Transition Click 1 – 100% | 99 | Transition Ramp Down Short Smooth 2 – 50 to 0% |
| 18 | Strong Click 2 - 80% | 59 | Transition Click 2 – 80% | 100 | Transition Ramp Down Long Sharp 1 – 50 to 0% |
| 19 | Strong Click 3 - 60% | 60 | Transition Click 3 – 60% | 101 | Transition Ramp Down Long Sharp 2 – 50 to 0% |
| 20 | Strong Click 4 - 30% | 61 | Transition Click 4 – 40% | 102 | Transition Ramp Down Medium Sharp 1 – 50 to 0% |
| 21 | Medium Click 1 - 100% | 62 | Transition Click 5 – 20% | 103 | Transition Ramp Down Medium Sharp 2 – 50 to 0% |
| 22 | Medium Click 2 - 80% | 63 | Transition Click 6 – 10% | 104 | Transition Ramp Down Short Sharp 1 – 50 to 0% |
| 23 | Medium Click 3 - 60% | 64 | Transition Hum 1 – 100% | 105 | Transition Ramp Down Short Sharp 2 – 50 to 0% |
| 24 | Sharp Tick 1 - 100% | 65 | Transition Hum 2 – 80% | 106 | Transition Ramp Up Long Smooth 1 – 0 to 50% |
| 25 | Sharp Tick 2 - 80% | 66 | Transition Hum 3 – 60% | 107 | Transition Ramp Up Long Smooth 2 – 0 to 50% |
| 26 | Sharp Tick 3 – 60% | 67 | Transition Hum 4 – 40% | 108 | Transition Ramp Up Medium Smooth 1 – 0 to 50% |
| 27 | Short Double Click Strong 1 – 100% | 68 | Transition Hum 5 – 20% | 109 | Transition Ramp Up Medium Smooth 2 – 0 to 50% |
| 28 | Short Double Click Strong 2 – 80% | 69 | Transition Hum 6 – 10% | 110 | Transition Ramp Up Short Smooth 1 – 0 to 50% |
| 29 | Short Double Click Strong 3 – 60% | 70 | Transition Ramp Down Long Smooth 1 – 100 to 0% | 111 | Transition Ramp Up Short Smooth 2 – 0 to 50% |
| 30 | Short Double Click Strong 4 – 30% | 71 | Transition Ramp Down Long Smooth 2 – 100 to 0% | 112 | Transition Ramp Up Long Sharp 1 – 0 to 50% |
| 31 | Short Double Click Medium 1 – 100% | 72 | Transition Ramp Down Medium Smooth 1 – 100 to 0% | 113 | Transition Ramp Up Long Sharp 2 – 0 to 50% |
| 32 | Short Double Click Medium 2 – 80% | 73 | Transition Ramp Down Medium Smooth 2 – 100 to 0% | 114 | Transition Ramp Up Medium Sharp 1 – 0 to 50% |
| 33 | Short Double Click Medium 3 – 60% | 74 | Transition Ramp Down Short Smooth 1 – 100 to 0% | 115 | Transition Ramp Up Medium Sharp 2 – 0 to 50% |

Figure B.1: Partial contents of the TouchSense 2200 library, taken from section 12.1.2 of [34].



| EFFECT ID NO. | WAVEFORM NAME | EFFECT ID NO> | WAVEFORM NAME | EFFECT ID NO. | WAVEFORM NAME |
|---|---|---|---|---|---|
| 34 | Short Double Sharp Tick 1 – 100% | 75 | Transition Ramp Down Short Smooth 2 – 100 to 0% | 116 | Transition Ramp Up Short Sharp 1 – 0 to 50% |
| 35 | Short Double Sharp Tick 2 – 80% | 76 | Transition Ramp Down Long Sharp 1 – 100 to 0% | 117 | Transition Ramp Up Short Sharp 2 – 0 to 50% |
| 36 | Short Double Sharp Tick 3 – 60% | 77 | Transition Ramp Down Long Sharp 2 – 100 to 0% | 118 | Long buzz for programmatic stopping – 100% |
| 37 | Long Double Sharp Click Strong 1 – 100% | 78 | Transition Ramp Down Medium Sharp 1 – 100 to 0% | 119 | Smooth Hum 1 (No kick or brake pulse) – 50% |
| 38 | Long Double Sharp Click Strong 2 – 80% | 79 | Transition Ramp Down Medium Sharp 2 – 100 to 0% | 120 | Smooth Hum 2 (No kick or brake pulse) – 40% |
| 39 | Long Double Sharp Click Strong 3 – 60% | 80 | Transition Ramp Down Short Sharp 1 – 100 to 0% | 121 | Smooth Hum 3 (No kick or brake pulse) – 30% |
| 40 | Long Double Sharp Click Strong 4 – 30% | 81 | Transition Ramp Down Short Sharp 2 – 100 to 0% | 122 | Smooth Hum 4 (No kick or brake pulse) – 20% |
| 41 | Long Double Sharp Click Medium 1 – 100% | 82 | Transition Ramp Up Long Smooth 1 – 0 to 100% | 123 | Smooth Hum 5 (No kick or brake pulse) – 10% |

Figure B.2: Part 2 of the contents of the TouchSense 2200 library, taken from section 12.1.2 of [34].

# C

# Latency measurement results

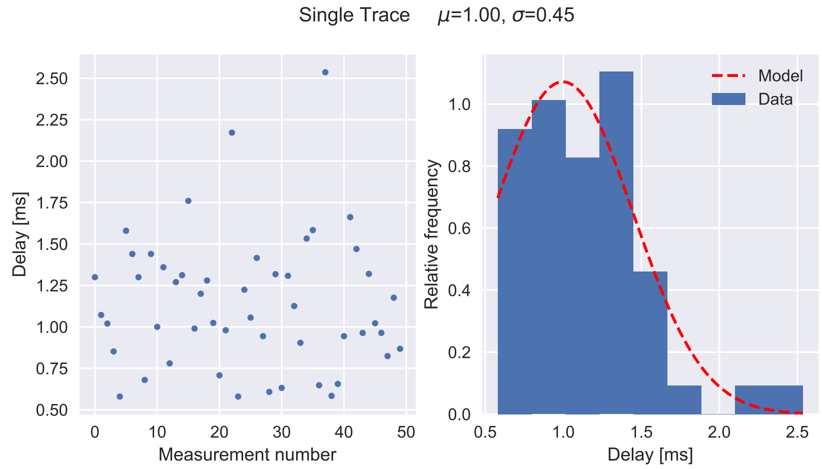


Figure C.1: Latency test with driver directly attached and in RTP mode.

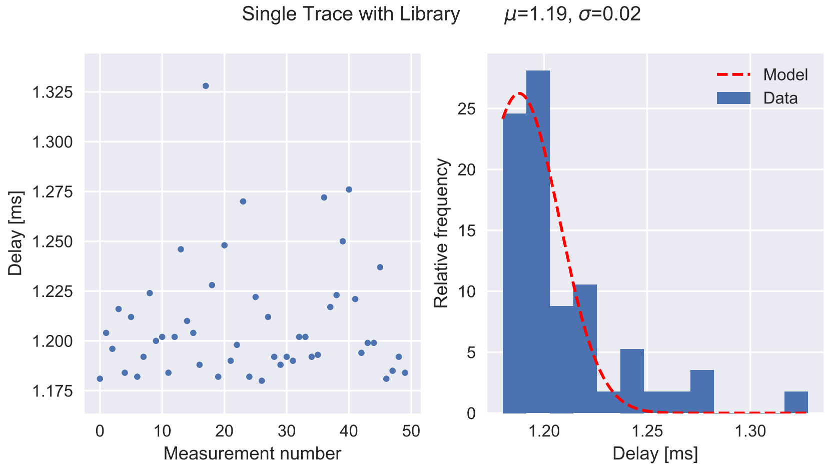


Figure C.2: Latency test with driver directly attached and using the waveform memory, triggered over $I^2C$.

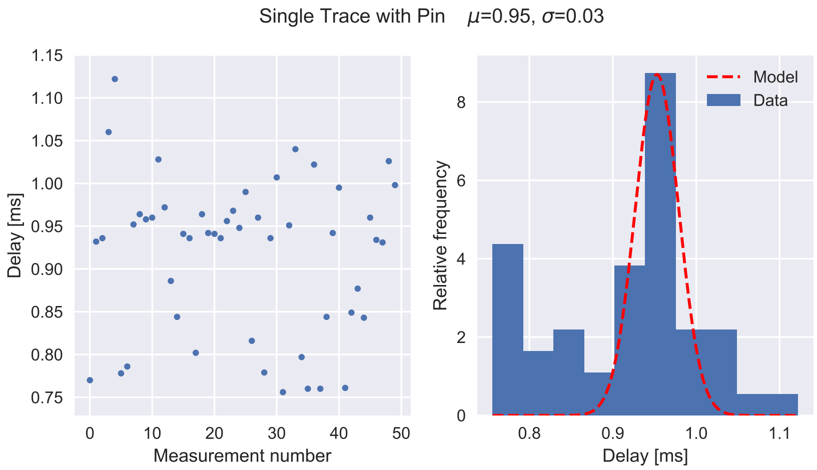


Figure C.3: Latency test with driver directly attached and using the waveform memory, triggered by the IN/TRIG pin.

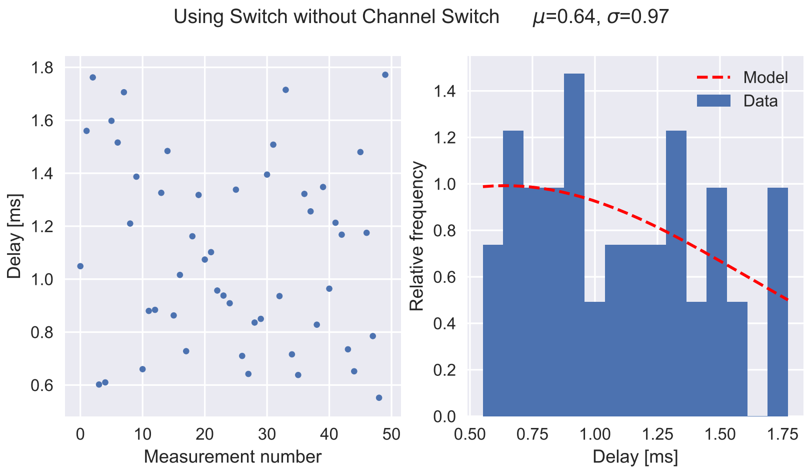


Figure C.4: Latency test with switch between the driver and the microcontroller without a switch in output channel, using RTP mode.

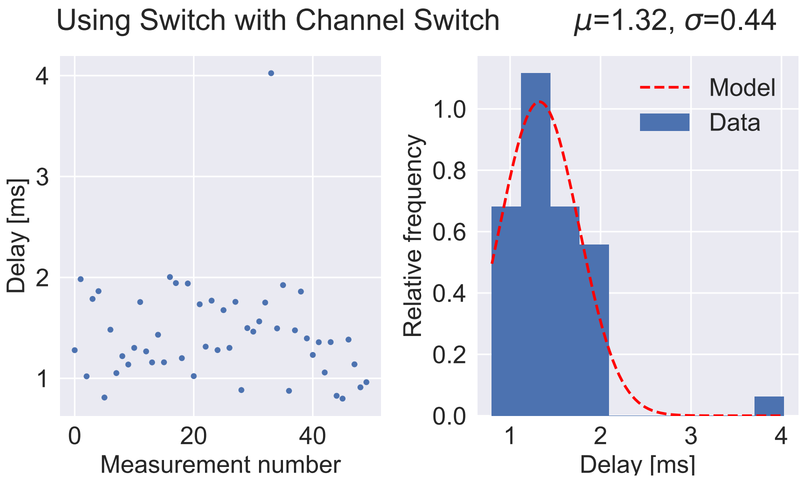


Figure C.5: Latency test with switch between the driver and the microcontroller with a switch in output channel, using RTP mode.

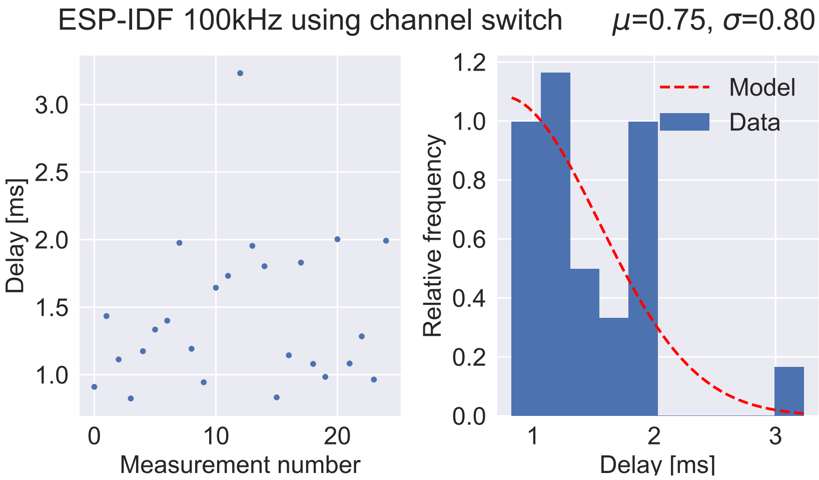


Figure C.6: Latency test on the first prototype PCB using the ESP-IDF framework with a 100 kHz I$^2$C bus, using RTP mode.

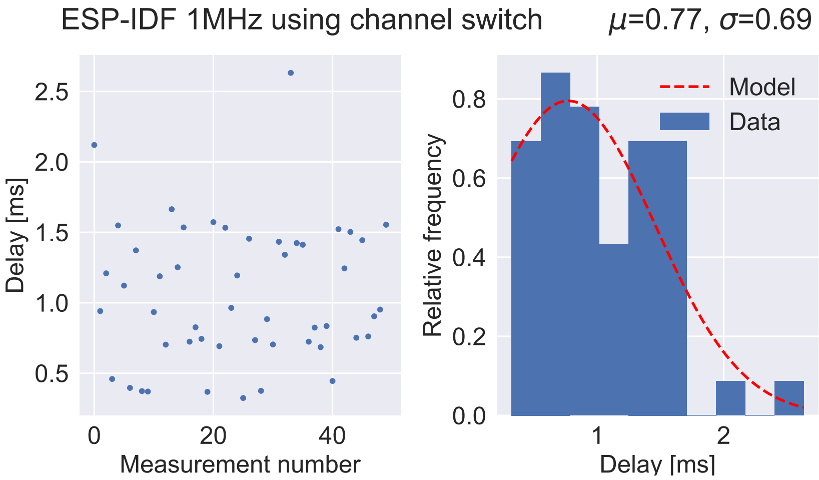


Figure C.7: Latency test on the first prototype PCB using the ESP-IDF framework with a 1 MHz I$^2$C bus, using RTP mode.

# D

# LRA characterization

## D.1. Test setup manufacturer

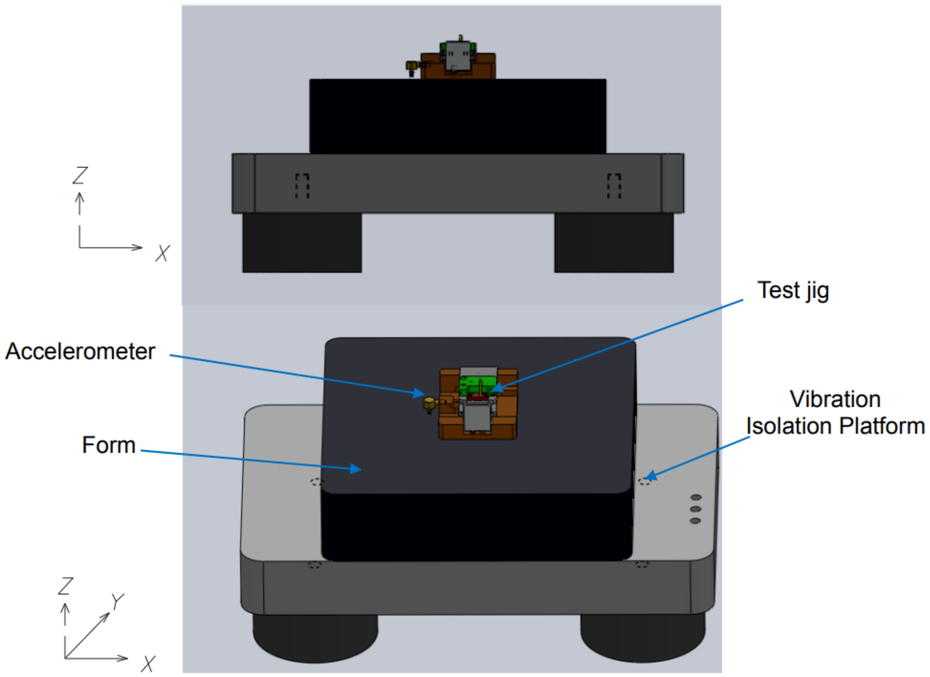


Figure D.1: Test setup used by the manufacturer of the LRAs, taken from [46].

Table D.1: The gains of the accelerometer at different orientations.

| x-gain [bits/G] | y-gain [bits/G] | z-gain [bits/G] |
|---|---|---|
| 263.3 ± 1.6 | 256 ± 1.0 | 251.0 ± 1.7 |

## D.2. Sine fits

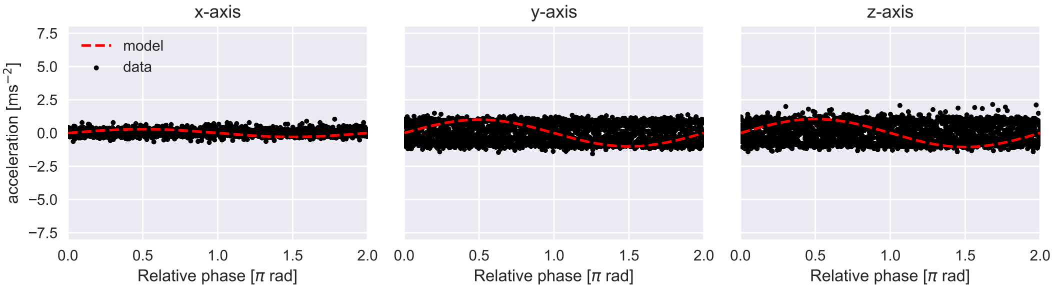


Figure D.2: Phase folded fit of the sine at 25% power. As the resonance tracking feature of the driver was unable to settle on a frequency, the graph could not properly be phase-folded.

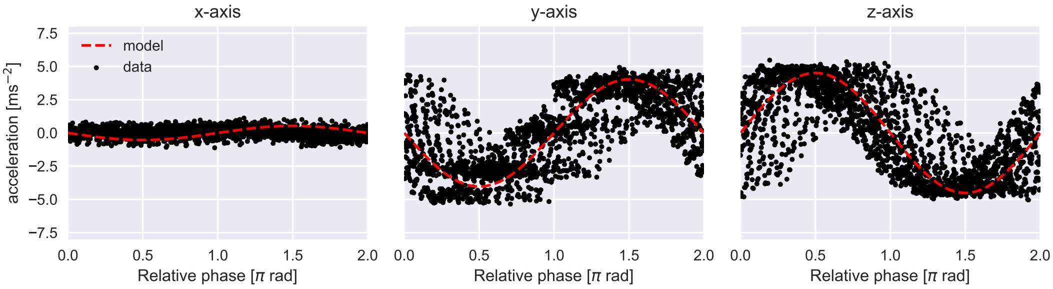


Figure D.3: Phase folded fit of the sine at 75% power. There is a drift in frequency showing in this graph.

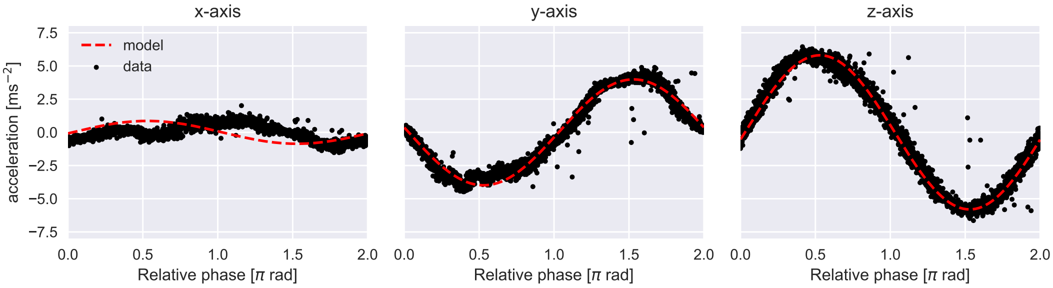


Figure D.4: Phase folded fit of the sine at 100% power.

## D.3. Envelope fits

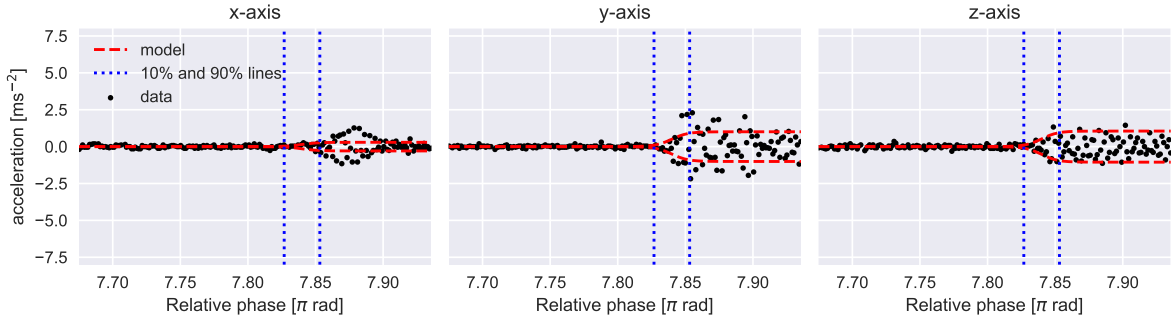


Figure D.5: Envelope fit to define the rise time at 25% power.

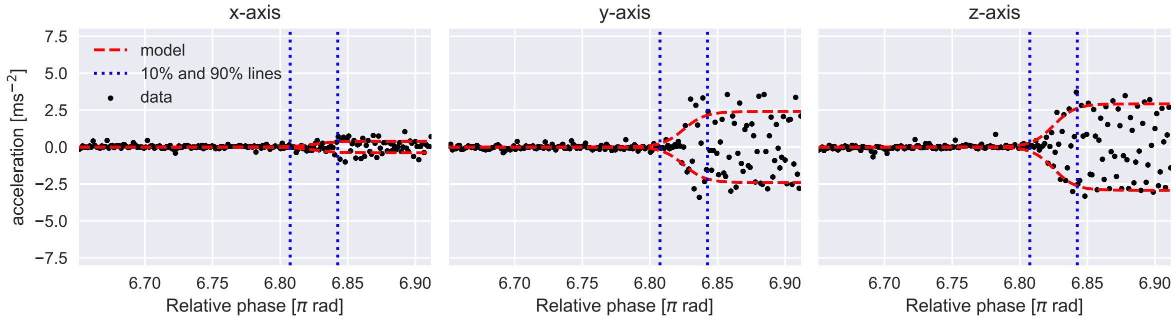


Figure D.6: Envelope fit to define the rise time at 50% power.

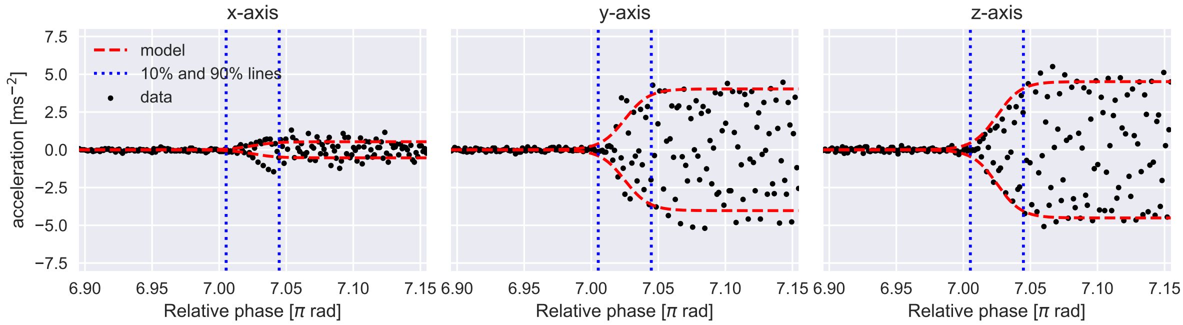


Figure D.7: Envelope fit to define the rise time at 75% power.

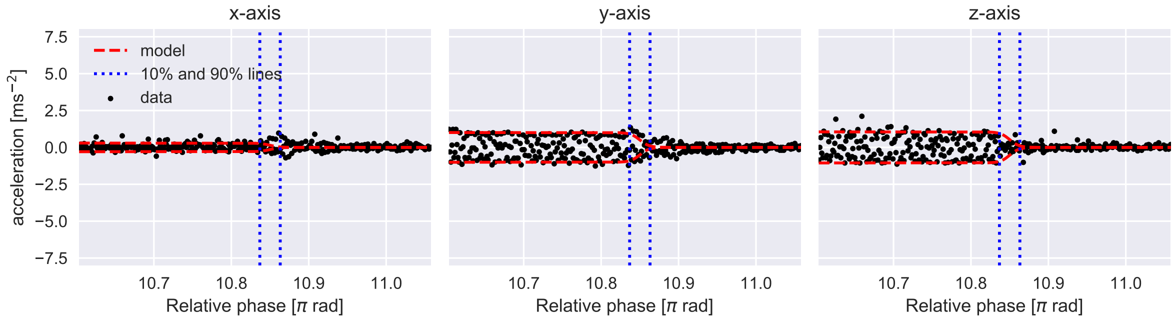


Figure D.8: Envelope fit to define the fall time at 25% power.

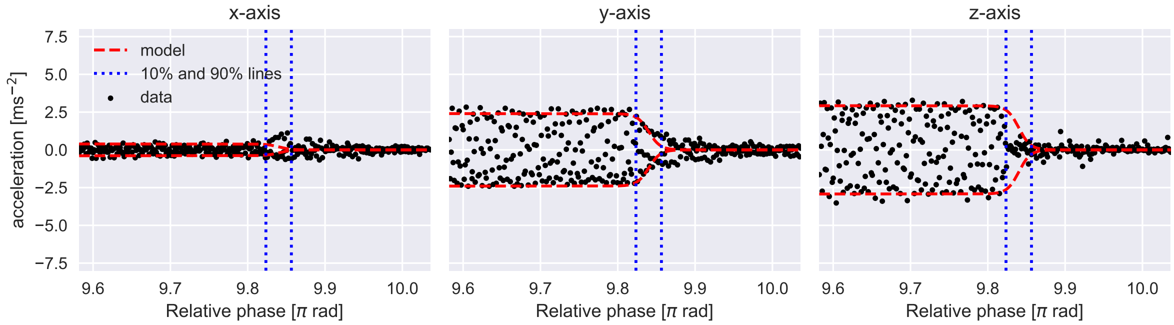


Figure D.9: Envelope fit to define the fall time at 50% power.

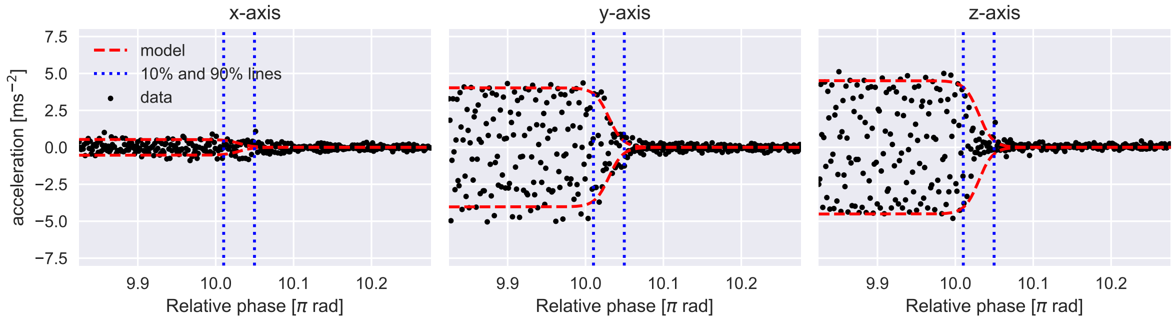


Figure D.10: Envelope fit to define the fall time at 75% power.

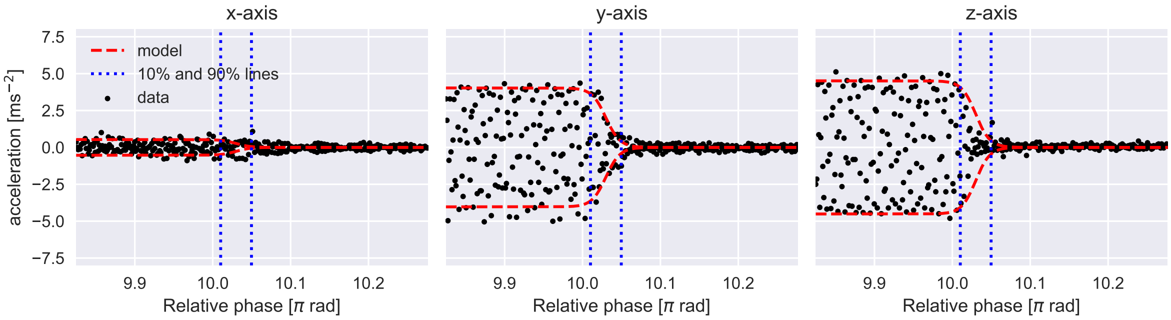


Figure D.11: Envelope fit to define the fall time at 100% power.

# Bibliography

[1] L. G. J. van der Knaap, "Softglove: Per finger force feedback," Bachelor thesis, Delft University of Technology, 2019.

[2] T. R. K. Peelen, "Softglove: Palm vibrotactile feedback," Bachelor thesis, Delft University of Technology, 2019.

[3] I. Evenden, "The history of virtual reality," 2016 (accessed 15-06-2019). [Online]. Available: https://www.sciencefocus.com/future-technology/the-history-of-virtual-reality/

[4] J. Lieberman and C. Breazeal, "Tikl: Development of a wearable vibrotactile feedback suit for improved human motor learning," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 919–926, 2007.

[5] C. Lam, "Sense glove: Dissertation rev. a," Master's thesis, The Hague University of Applied Sciences, The Netherlands, 2018.

[6] M. Corten, "Sense glove haptics research 2018: Human machine interface optimization," 2018.

[7] J. Perret and E. Vander Poorten, "Touching virtual reality: a review of haptic gloves," June 2018, p. 7.

[8] C. Pacchierotti, S. Sinclair, M. Solazzi, A. Frisoli, V. Hayward, and D. Prattichizzo, "Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives," *IEEE transactions on haptics*, vol. 10, no. 4, pp. 580–600, 2017.

[9] EuroCircuits, 2019. [Online]. Available: https://www.eurocircuits.com/

[10] J. Fjelstad, *Flexible Circuit Technology, Third Edition.* Br Publishing, Incorporated, 2007.

[11] C. Lam, "Senseglove communication protocol," 2018.

[12] P. V. den Bossche, F. Vergels, J. V. Mierlo, J. Matheys, and W. V. Autenboer, "Subat: An assessment of sustainable battery technology," *Journal of Power Sources*, vol. 162, no. 2, pp. 913 – 919, 2006, special issue including selected papers from the International Power Sources Symposium 2005 together with regular papers. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378775305008761

[13] G. E. Blomgren, "Current status of lithium ion and lithium polymer secondary batteries," in *Fifteenth Annual Battery Conference on Applications and Advances (Cat. No. 00TH8490).* IEEE, 2000, pp. 97–100.

[14] T. Instruments, 2019. [Online]. Available: https://www.ti.com/packaging/docs/searchtipackages.tsp?packageName=BGA

[15] Distrelec, 2019. [Online]. Available: https://www.distrelec.nl/nl/converter-ic-qfn-24-ic-haus-ic-tw4-qfn24/p/17350861

[16] H. Huang, T. Li, C. Antfolk, C. Enz, J. Justiz, and V. M. Koch, "Experiment and investigation of two types of vibrotactile devices," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob).* Ieee, 2016, pp. 1266–1271.

[17] D. A. Kontarinis and R. D. Howe, "Tactile display of vibratory information in teleoperation and virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 4, no. 4, pp. 387–402, 1995.

[18] K. Takahashi, J. Saiki, and K. Watanabe, "Realignment of temporal simultaneity between vision and touch," *Neuroreport*, vol. 19, no. 3, pp. 319–322, 2008.

[19] A. Wilska, "On the vibrational sensitivity in different regions of the body surface," *Acta Physiologica Scandinavica*, vol. 31, no. 2-3, pp. 285–289, 1954.

[20] F. Gonzalez, F. Gosselin, and W. Bachta, "A framework for the classification of dexterous haptic interfaces based on the identification of the most frequently used hand contact areas," in *2013 World Haptics Conference (WHC)*. IEEE, 2013, pp. 461–466.

[21] M. Bannwart, P. Pyk, D. Kiper, K. Eng, R. Gassert, and Y. Kim, "Usability assessment of low-cost vibration motors for presenting vibrotactile feedback in sensory and motor rehabilitation," in *2013 International Conference on Virtual Rehabilitation (ICVR)*. IEEE, 2013, pp. 206–207.

[22] M. Azadi and L. A. Jones, "Vibrotactile actuators: Effect of load and body site on performance," in *2014 IEEE Haptics Symposium (HAPTICS)*. IEEE, 2014, pp. 351–356.

[23] C. Seim, J. Hallam, S. Raghu, T.-A. Le, G. Bishop, and T. Starner, "Perception in hand-worn haptics: Placement, simultaneous stimuli, and vibration motor comparisons," Georgia Institute of Technology, Tech. Rep., 2015.

[24] P. Microdrives, "Adding and improving haptics," 2016 (accessed 16-06-2019). [Online]. Available: https://www.precisionmicrodrives.com/haptic-feedback/adding-and-improving-haptics/

[25] P. Wellman and R. D. Howe, "Towards realistic vibrotactile display in virtual environments," in *Proceedings of the ASME Dynamic Systems and Control Division*, vol. 57, no. 2, 1995, pp. 713–718.

[26] P. Microdrives, "Linear resonant actuators - lra," 2015 (accessed 16-06-2019). [Online]. Available: https://www.precisionmicrodrives.com/vibration-motors/linear-resonant-actuators-lras/

[27] *G0832022D LRA Coin Type Vibration Motor*, Jinlong Machinery & Electronics, September 2017, rev. 1.

[28] *C1020B217F ERM Coin Type Vibration Motor*, Jinlong Machinery & Electronics, March 2013, rev. 0.

[29] F. Wang, "Haptic energy consumption," May 2014. [Online]. Available: http://www.ti.com/lit/an/sloa194/sloa194.pdf

[30] M. Kato and K. Hirata, "Characteristic evaluation of linear resonant actuator utilizing electrical resonance," *IEEJ Journal of Industry Applications*, vol. 7, no. 2, pp. 175–180, 2018.

[31] A. Yoshitake, K. Harada, T. Todaka, Y. Ishihara, and K. Hirata, "Dynamic analysis of a linear oscillatory actuator under feedback control," *IEEE Transactions on Magnetics*, vol. 33, no. 2, pp. 1662–1665, 1997.

[32] B. Burk, "Trouble with the drv2603evm-ct," 2012 (accessed 22-5-2019). [Online]. Available: http://e2e.ti.com/support/motor-drivers/f/38/p/223020/795834#795834

[33] *DRV2604L 2- to 5.2-V Haptic Driver for LRA and ERM with Internal Memory and Smart-Loop Architecture*, Texas Instruments, March 2018, rev. F.

[34] *DRV2605L 2- to 5.2-V Haptic Driver for LRA and ERM with Effect Library and Smart-Loop Architecture*, Texas Instruments, March 2018, rev. D.

[35] K. Griffin, "Drv2605l: Diagnostics and auto calibration modes both fail," 2018 (accessed 16-6-2019). [Online]. Available: http://e2e.ti.com/support/motor-drivers/f/38/p/673081/2478637#2478637

[36] *Single I2C/SMBus Address Translator*, Analog Devices, October 2015, rev. A.

[37] *TCA9548A Low-Voltage 8-Channel $I^2C$ Switch with Reset*, Texas Instruments, November 2016, rev. F.

[38] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

[39] *ESP8266EX Datasheet*, Espressif Systems, November 2018, rev. 6.0.

[40] *ESP32 Series Datasheet*, Espressif Systems, April 2019, rev. 3.0.

[41] *ADXL345 3-axis Digital Accelerometer*, Analog Devices, June 2015, rev. E.

[42] L. Fried, "Adafruit drv2605 library," 2016. [Online]. Available: https://github.com/adafruit/Adafruit_DRV2605_Library

[43] C. Jay, M. Glencross, and R. Hubbold, “Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment,” *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 14, no. 2, p. 8, 2007.

[44] M. Meehan, S. Razzaque, M. C. Whitton, and F. P. Brooks, “Effect of latency on presence in stressful virtual environments,” in *IEEE Virtual Reality, 2003. Proceedings.* IEEE, 2003, pp. 141–148.

[45] P. Microdrives, “Linear resonant actuators - lras,” 2018 (accessed 21-6-2019). [Online]. Available: https://www.precisionmicrodrives.com/vibration-motors/linear-resonant-actuators-lras/

[46] *LV081538-GO1 LRA Linear Vibration Motor*, Jinlong Machinery & Electronics, March 2019, rev. 5.